REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT DATE	3. DATES COVERED *(From - To)*
23-07-2003	Fifth Annual Report	15/08/01 - 30/11/02

4. TITLE AND SUBTITLE

1997 MURI in RF Photonics: RF Photonics for Array Processing

5a. CONTRACT NUMBER

5b. GRANT NUMBER
MURI 1997 GN00014-97-1-1006

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Kelvin H. Wagner, University of Colorado, Boulder
Dana Anderson, University of Colorado, Boulder
Zoya Popovic, University of Colorado, Boulder
Randall W. Babbitt, Montana State University
Andre Knoesen, University of California, Davis
Lloyd Griffiths, George Mason University

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Colorado,
Optoelectronic Computing Systems Center,
Campus Box 525
Boulder, CO 80309

8. PERFORMING ORGANIZATION REPORT NUMBER

153 6702

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr William Miceli,
Office of Naval Research, Code 313,
Ballston Tower #1, Rm 619,
Arlington, VA 22217

Dr. Keith Williams
Naval Research Laboratory
4555 Overlook Ave. SW
Washington DC 20375-5338

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

12. DISTRIBUTION AVAILABILITY STATEMENT

Unlimited

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20030807 102

13. SUPPLEMENTARY NOTES
Funded by OSD DDR&E

14. ABSTRACT

This report covers the fifth year of progress of the 1997 MURI on RF Photonics for Antenna Arrays at the University of Colorado, Montana State University, George Mason University, and the University of California Davis. Novel techniques for optical control and processing of the wideband RF and microwave signals encountered in phased array antennas have been developed, guided by research in spatio-temporal adaptive processing algorithms and active quasi-optical RF antenna arrays. The primary goal of this research is to develop enabling optical techniques that provide dramatic improvements in antenna array performance over conventional RF, optical, and digital techniques, allowing the efficient processing of large broadband antenna arrays. Coherent modulation and detection is made robust and practical by the use of dynamic holography in photorefractive and optical coherent transient media. This report summarizes the teams management, educational, and outreach activities, as well as technical progress in the fifth year on the constituent projects - broadband adaptive optical array processing, spatio-temporal array-processing algorithms, coherent-transient true-time-delay, photorefractive signal extraction, optical antenna control, and polymer in-line fiber modulators.

15. SUBJECT TERMS

Phased Array Antennas, coherent transients, photorefractive, RF photonics, STAP processing

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	Unlimited - UL	381	
Unclassified	Unclassified	Unclassified			19b. TELEPONE NUMBER *(Include area code)*

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI-Std Z39-18

RF Photonics for Array Processing MURI 5th year summary

Attention: **George D. McNeal**,
ONR 313, 800 N Quincy St., Arlington, VA 22217-5660

Principal Investigator: **Kelvin Wagner**, *CU-ECE*
Dept. of ECE and OCS, Box 425, University of Colorado, Boulder CO, 80309-0425
kelvin@optics.colorado.edu, (303)-492-4661 (5810 FAX)
http://optics.colorado.edu/MURI

University of Colorado Co-PI:	**Dana Anderson**,	*CU-Physics*
University of Colorado Co-PI:	**Zoya Popović**,	*CU-ECE*
George Mason University Co-PI:	**Lloyd Griffiths**,	*GMU-EE*
Montana State University Co-PI:	**Randall Babbitt**,	*MSU-Phys*
University of California at Davis Co-PI:	**Andre Knoesen**,	*UCD-ECE*

1   5th Year Executive Summary

This report summarizes the fifth year of the MURI in RF Photonic Systems which assembles the University of Colorado, the University of California-Davis, Montana State University, and George Mason University into a systems oriented research team investigating the application of photonic techniques to the control and processing of RF phased arrays. Under this MURI funding, the prime contract is administered through the University of Colorado under the direction of Prof. Kelvin Wagner.

Photonic techniques have emerged as the preferred approach to RF communication and remoting tasks throughout the microwave/millimeter-wave bands, when size, weight, power, and dispersion free bandwidth are key factors. In addition, applications such as controlling time delay, spectral filtering, mixing, frequency generation, antenna array control, beam forming, wide-band signal processing, and target recognition are being implemented optically. Much of this development has been driven by the evolution of high frequency photonic devices such as 100 GHz detectors, 40 GHz modulators, 20 GHz lasers, and 3 THz fiber optic transmission and delay lines. Often these technologies have been used as direct replacement

of individual RF components with their optical counterparts, relying on the smaller size and lighter weight of the optical components and waveguides to yield a resulting system advantage. More sophisticated applications utilize the incredibly large bandwidth of optical transmission to simultaneously wavelength-multiplex a large number of RF signals for efficient parallel transmission or to control the element time delays through dispersion. The approach of this MURI team takes RF photonics to an even higher level of sophistication, in which the massive parallelism possible using dynamic volume holography in photorefractive and photon-echo materials enables the implementation of optimal adaptive and nonlinear algorithms for array processing in the optical domain.

The system research groups comprising this MURI team have pursued several different avenues towards one common goal of inserting photonics into RF array systems. We have developed solutions that build on the strengths of each approach, and identified the optimal combination of devices, materials, algorithms, and systems. The BEAMTAP approach (Broadband and Efficient Adaptive Method for True-time-delay Array Processing) has been investigated theoretically and implemented experimentally to demonstrate efficient adaptive beam forming using acousto-optic deflectors, traveling-fringes detectors, and photorefractive adaptive weights. The novel physics of optical coherent transients (OCT) have been developed for applications as programmable RF time delays as well as array signal processors, since these nonlinear materials can directly produce the necessary delayed RF signals as photon echoes. Both of these approaches utilize coherent beam forming in the optical domain which can improve the noise figure of optically remoted arrays and optical novelty filtering developed at CU can enable robust RF signal combination even in the presence of phase errors and drifts. Optically controlled quasi-optical active antenna arrays provide a near term test vehicle for optically controlled arrays, allowing rapid switching between polarizations, transmit/receive (T/R) mode, and frequency response, and in addition, demonstrate the capability of producing multiple simultaneous true-time-delay (TTD) beams as a front end for beam-space adaptive optical processors. Optical nonlinearities have been explored as an approach to dynamically solving the problem of signal extraction from small RF antenna arrays. Practical polymeric in-line fiber (PILF) modulators with high-speed coplanar waveguide (CPW) electrode structures have been developed as the front end RF-to-optical transducers for the antenna arrays. All of these efforts have been tied together by investigations of spatio-temporal signal processing algorithms compatible with RF photonic technology, which allows us to evaluate and compare the various optical systems techniques being investigated. As an example system developed as a unique collaboration between MURI team members, an RF antenna array has been coupled into a photorefractive optical system for extracting the principle component in the RF signal environment, and the resulting system has been packaged as a portable suitcase demonstrator.

Contents

2 Education, Presentations, Outreach, and Management

2.1 MURI Team

This MURI effort has directly or indirectly funded the 6 principal investigators, 5 post-doctoral or visiting research scientists, 17 graduate students, 11 undergraduate students, 2 research assistants, and fractions of a financial manager and secretary.

2.1.1 Principal Investigators

The original MURI team of 6 PIs was joined by Andre Knoesen from UC Davis in year 2 year, replacing Ted Weverka. Prof. Andre Knoesen worked on PILF modulators on a seperately funded subcontract which ends this year. Weverka's project on coherent optical beam combination for phased array processing is being continued at the University of Colorado as the required front end to the BEAMTAP processor.

Table 1: Principal investigators involved in the RF Photonic Systems program.

Researcher	Department	Institution
Prof. Kelvin Wagner	ECE/OCS	CU-Boulder
Prof. Zoya Popović	ECE	CU-Boulder
Prof. Dana Anderson	JILA	CU-Boulder
Prof. William Randall Babbitt	Physics	Montana State University
Dean Lloyd Griffiths	ITE	George Mason University
Prof. Andre Knoesen	ECE	UC-Davis

2.2 Students, Post-Doctoral Researchers, and other Personnel

Table 2: Undergraduate students who have been involved in MURI related research during the past year. Again, those who worked on closely related projects in collaboration with the MURI effort are indicated by *.

Undergrad	Status	Position	Advisor	Dept.	Inst.
Eric Hoyt	graduated BS/MS 12/02	Undergraduate	Wagner	OCS	CU
John Jost	Assistant	Undergraduate	Anderson	Phys	CU
Joe Fischer*	graduated 6/02	Undergrad	Babbitt	Physics	MSU
Zeb Barber	graduated 6/03	Undergrad	Babbitt	Physics	MSU
Jesse Law	continuing	Undergrad	Babbitt	Physics	MSU

Table 3: MURI personnel indicating those funded by this program. Those who worked on closely related projects in collaboration with the MURI effort are indicated by *.

Researcher	Status	Position	Advisor	Dept.	Inst.
Andrew Kiruluta	6/02-11/02	Post-doc	Wagner	OCS	CU
Steve Jia	continuing	Grad student	Babbitt	Physics	MSU
Mingzhen Tian	continuing	Research Scientist	Babbitt	Physics	MSU
Diego Yankelevich	3/99-present	Adjunct Prof.	Knoesen	EE	Davis
Joe Shamir	9/02-10/02	Visiting MURI fellow	Wagner	OCS	CU
Edilene Fotheringham	continuing	Grad Student	Anderson	JILA	CU
Greg Kriehn	DoD fellowship	Grad Student	Wagner	OCS	CU
Friso Schlottau	continuing	Grad Student	Wagner	OCS	CU
Paul Smith*	continuing	Grad Student	Popović	ECE	CU
Jan Peeters Weem	PhD finished 6/02	Grad Student	Popović	ECE	CU
Jacques Hong Loui	continuing	Grad Student	Popović	ECE	CU
Bob Peters*	MS finished 8/01	Grad Student	Babbitt	Physics	MSU
Randy Reibel	Graduated 8/02	Grad Student	Babbitt	Physics	MSU
Hongyan Li	continuing	Grad Student	Babbitt	Physics	MSU
John Campbell	continuing	Grad student	Knoesen	EE	Davis
Liu Ming Wu	continuing	Grad student	Knoesen	EE	Davis
Leslie Czaia	continuing	Prof. Res. Asst.	Anderson	JILA	CU
John Getty		EE engineer	Babbit	Physics	MSU
Norm Williams		Machinist	Babbit	Physics	MSU

2.2.1 Students Graduated from the MURI program

1. Jan Peeters Weem, June 2002, CU, ECE, PhD, Popović. Thesis title: "Broadband antenna arrays and noise coupling for radio astronomy."

2. Randy Reibel, August 2002, MSU, Physics, PhD, Babbit.

3. Edeline Fotheringham, May 2003, CU, ECE, PhD, Anderson. Thesis title: "Microwave signal processing with photorefractive dynamic holography."

4. Greg Kriehn, May 2003, CU, ECE, PhD, Wagner. Thesis title: "Coherent optical signal processing for broadband adaptive phased-array antennas using the BEAMTAP algorithm."

2.3 Meetings

The Boulder team met weekly to exchange information, make presentations, as well as to discuss technical and management details. The entire MURI team gathers once or twice a year for intellectual exchange, discussions, and review regarding scientific progress during the year. One of these meetings is primarily an internal MURI meeting, and the other is a formal review for programmatic purposes. Our third annual review was held in Boulder on October 12 2000 conjunction with the UCLA MURI review on October 13. Plans for the 4th annual review to be held in Washington DC on Nov 30, 2001 were put on hold when the Program Manager, Dr. Steve Pappert of SPAWAR, took a leave from the government.

2.3.1 5th year MURI team Interactions

1. Prof. Popović's group has collaborated mainly with Prof. Anderson's group at CU, and somewhat with Prof. Griffiths at MSU.

2. Prof. Popović visited Prof. Knoessen's group at UC Davis and gave a graduate seminar, October 2002.

3. Prof. Babbitt visited UC-Boulder in November of 2002.

2.4 Additional Collaborations and Related Research

The MURI has enabled a wide variety of collaborations among the PIs as well as with other groups. In this section some notable collaborations are discussed and outgrowths from the MURI into new areas are mentioned.

1. Prof. Joseph Shamir of the Technion in Israel was the MURI fellow and visited Prof. Wagner's group between Aug. and Sept. of 2001 and between Sept. and Oct. of 2002.

2. Prof. Vitaly Voloshinov of Moscow State University in Russia was the MURI fellowship visitor to Prof. Wagner's group to work on acousto-optic devices, Nov 11-21, 2001.

3. Prof. Babbit explored Raman amplification at 793nm for OCTs with Directed Energy Systems of Colorado Springs in the Fall of 2002.

4. Two graduate students under the supervision of Prof. Popović have been for 3 months at the NFRA facility at Dwingeloo (Jan Peeters Weem and Eric Bryerton). Several papers have resulted from this collaboration.

5. Technishe Universität München (Germany) and Univ. of Karlsruhe (Germany): Two German students, Holger Matern and Johannes Russer were sent (and supported by their home country) to work in Prof. Popović's group on their MS theses in the area of RF photonics. Holger developed a 5-channel electrooptic modulator and Johannes is finishing a fast optically-controlled microwave phase shifter.

2.4.1 Other related grants

1. Profs. Popović and Wagner submitted new proposals on optical processing of RF signals from broadband RF antenna arrays. Any subsequent grant will be a direct result of the Co-PI's joint work under this MURI.

2. Prof. Popovic and Anderson have received a follow-on grant from NSF to expand the optical processing to wireless communications. This grant is now ending, and they submitted a follow-on proposal to the NSF ITP program, Dec. 2002. The proposal is currently under review.

3. "High Performance Correlators based on Spectral Hole Burning Technology," NASA, PI: Alan Craig, $5,096,000, 4/99-4/03. PI changed to W. R. Babbitt in 12/2001.

4. Prof. Babbit received an award from Montana Board for Research and Commercialization of Technology: matching grant for the MURI grant $7,810, 5/01-5/02.

5. "Consortium for Stable Laser Applications," NSF Partners for Innovation (PFI), PI: Alan Craig, co-PI Jim McMillan, $598,553, 10/01-9/03, changed PI to W. R. Babbitt in 12/01.

6. "Applications of Optical Coherent Transient Technology: Pulse Shaping, Spectral Filtering, Arbitrary Waveform Generation, and RF Beamforming," AFOSR/ DEPSCOR, PI: W. R. Babbitt, co-PI: Mingzhen Tian, $310,067 plus $155,034 match, 7/02-7/05.

7. "Wide-Bandwidth and Multidimensional Analog Signal Processing," DARPA, Lead: W. R. Babbitt, Co-PIs Kelvin Wagner, Univ. of Colorado and Randy Equall, Scientific Materials Corp., and Kris Merkel, Krishna Rupavatharam, and Rufus Cone at MSU, $1.9M, 10/02-9/04.

8. "Spatial Spectral Coherent Holographic Integrating Processor-S2-CHIP: Device characterization, performance evaluation, and optimization," Montana Board for Research and Commercialization of Technology, PI: W. R. Babbitt, co-PIs: Kris Merkel and Mark Ivey,$149,420, 11/02-7/03, matched by NASA grant.

9. Prof. Wagner, $60K subcontract to University Nebraska Lincoln for ONR proposal on Random Noise Radar. Title: Acoustooptic Delay Line/Correlator for random noise radar, 8/00 - 8/02.

10. $350K Prof. Wagner PI, Real-time Multibeam Imaging and High-Resolution Spectral Analysis for Large, Wideband Antenna Arrays – submitted to NRO DII (National Reconnaissance Office Directors Innovation Initiative). $350,000 for 9 months. Status: Awarded on Dec 21, 2001, rescinded award due to NRO budget cut.

11. $20K Prof. Wagner PI, Joint Optoelectronic Project Proposal – Double-clad Thulium doped fiber for use in a fiber amplifier for optical coherent transient experiments. An equipment proposal to the Joint Optoelectronics Project (JOP). Status AWARDED: $20,000 in custom optical fiber

12. $14.5K Prof. Wagner PI, NSF/CNRS collaboration, multidimensional photon echo optical processing, notice of award Jan 3 2000, in effect 2000-2004.

13. Prof. Wagner, co-PI, $80K DEPSCOR subcontract from MSU, Optical coherent-transient time-delay control using acousto-optic distributed local oscillators, $80K, plus $40K CU matching 7/2002-12/2004.

14. Prof. Wagner PI, Coherent Optical Array Processing, DARPA Analog Optical Signal Processing (AOSP) program, awarded in 2002. In effect 2002-2006.

2.5   Awards and Honors for the PIs and students

1. Best Poster Award granted to Friso Schlottau, Randy Babbit, Jean-Louis Le Gouet and Kelvin Wagner for the CPIA poster entitled "Spatial spectral holography for wideband spectral analysis," Nov. 2002.

2. MWP 2001 on Jan. 2002. Best student presentation runner-up: Friso Schlottau and Kelvin Wagner, "RF Photonics for Simultaneous Multiple TTD Beamforming for 2-D Antenna Arrays," IEEE Microwave Photonics Meeting, Long Beach, Oct 2001 (delayed to Jan. 2002).

3. Prof. Kelvin Wagner was elevated to the status of fellow of the OSA (Optical Society of America) for "Contributions to adaptive spatio-temporal optical processing systems."

4. Zoya Popović and Dana Anderson both received the prestiguous Humboldt Award for Senior US Scientists from the German Alexander von Humboldt Stiftung (2001).

5. Zoya Popović received the HP/ASEE Terman Award for combined teaching and research excellence, October 2001.

6. Zoya Popović became a Fellow of the IEEE.

7. Zoya Popović received the Outstanding Speaker Award at the ONR Workshop on Challenges for Multifunctional Digital Transmit Arrays, Marco Island, Nov. 2001.

8. Zoya Popović received the CU-Subaru Educator Spotlight Award, 2003.

9. Prof. Babbit was awarded the Charles and Nora L. Wiley Faculty Award for Meritorious Research and Creativity, 2002.

10. Ph.D. student Randy Reibel received MSU's Graduate Achievement Award, 2003.

11. Undergraduate student Zeb Barber was nominated a Goldwater Scholar (2002) and graduated with Highest Honors (6/2003).

12. Randy Babbitt, became Director of Spectrum Lab, February 2002-present.

13. Committee chairs for Randy Babbitt: Chair of program subcommittee, Holography, Wavemixing, Photorefractives, and Storage, Conference on Lasers and Electro-Optics, Long Beach, CA, May 2002. Chair of program subcommittee, Holography, Wavemixing, Photorefractives, and Storage, Conference on Lasers and Electro-Optics, Baltimore, MD, June 2003. Co-Chair, Spectral Holeburning, Single Molecules, and Related Spectroscopies 2003, Bozeman, MT, July 2003.

2.6 MURI Publications

2.6.1 Papers published in 2002 (since 8/15/01)

1. Andrew Kiruluta, G. S. Pati, Gregory Kriehn, Paulo E. X. Silveira, Anthony W. Sarto and Kelvin Wagner, "Spatio-temporal operator formalism for holographic recording

and diffraction in a photorefractive based true-time-delay phased array processor," accepted for publication in Applied Optics, Sept. 2003.

2. J. Shamir and K. Wagner, "Generalized Bragg selectivity in volume holography," Applied Optics 41(32), pp. 6773-6785, Nov. 2002.

3. Paulo E. X. Silveira, G. S. Pati and Kelvin H. Wagner, "Optical Finite Impulse Response Neural Networks," Applied Optics, vol. 41, no. 20, pp. 4162-4180, 2002.

4. Steve Blair and Kelvin Wagner, Gated Logic with Optical Solitons, in "Collision-Based Computing", ed by Andrew Adamatzky, Springer-Verlag, UK, to be published in 2002.

5. Paulo E. X. Silveira, G. S. Pati and Kelvin H. Wagner "Optoelectronic implementation of a 256-channel sonar adaptive array processor," to be submitted to Applied Optics, 2003.

6. D. Z. Anderson, V. Damiao, D. Popovic, Z. Popovic, S. Romisch, A. Sullivan, "-70dB optical carrier suppression by two-beam coupling in photorefractive media," Applied Physics B, 72, pp. 743-748, 2001

7. E. Fotheringham, S. Romisch, P. Smith, D. Popovic, D. Anderson and Z. Popovic, "A lens antenna array with adaptive optical processing," IEEE Trans. Antennas and Propagation, Special Issue on Wireless Communications, vol. 50, no. 5, pp. 607-617 , May 2002.

8. H. Loui, J. Peeters Weem, Z. Popovic, "Dual-band dual-polarized nested Vivaldi slot array with multi-level ground plane," to appear in IEEE Trans. on Antennas and Propagation, 2003.

9. J. Peeters Weem, Z. Popovic, "Noise coupling in active phased array antennas," to be submitted to the IEEE Trans. on Microwave Theory and Techniques.

10. P. Smith, E. Fotheringham, Z. Popovic, D. Anderson, "An optoelectronic implementation of independent component analysis," to be submitted to IEEE Trans. on Circuits and Systems, August 2003.

11. R. Reibel, Z. Barber, M. Tian, and W. R. Babbitt, "High bandwidth spectral grating programmed with linear frequency chirps," J. Lumin. 98, 355-365 (2002).

12. Z. Barber, M. Tian, R. Reibel, and W. R. Babbitt, "Optical pulse shaping using optical coherent transients," Opt. Exp. 10, 1145-1150 (2002)

13. R. Reibel, Z. Barber, M. Tian, W.R. Babbitt, Z. Cole, and K.D. Merkel, "Amplification of high bandwidth phase modulated signals at 793nm," J. Opt. Soc. Am. B. 19, pp. 2315 (2002).

14. R. Reibel, Z. Barber, J. Fischer, M. Tian, and W. R. Babbitt, "High Bandwidth Linear Sideband Chirped Programming for Optical Coherent Transients," to be published in 2003.

15. J. Campbell, A. Knoesen and D. R. Yankelevich, "Measurement of the modulation efficiency of an optical phase-modulator using a self-homodyne receiver," IEEE Phot. Tech. Letters, vol. 14, no. 9, pp. 1130–1132, 2002.

2.6.2 Conference Presentations during 2002 (since 8/15/01)

1. G. Kriehn and K. Wagner, "Experimental Demonstration of Adaptive Phased-Array Signal Processing," (poster) *Colorado Photonics Industry Association*, November 2002.
2. G. Kriehn, F. Schlottau, and K. Wagner, "Optically-Implemented 2-D Beam Steering and Jammer Nulling using BEAMTAP," *Optical Computing*, April 2002.
3. G. Kriehn, F. Schlottau, G. S. Pati, and K. Wagner, "Demonstration of RF Photonic Beam Forming using the BEAMTAP Algorithm," *General Electric Meeting*, November 2001.
4. G. Kriehn and K. Wagner, "Experimental Adaptive Beam Forming with Polarization Read-Write Multiplexing using BEAMTAP," (poster) In *The International Topical Meeting on Microwave Photonics*, IEEE, Jan. 2002.
5. G. Kriehn, "Optical Phased Array Signal Processing," (poster) *Colorado Photonics Industry Association*, November 2001.
6. G. Kriehn and K. Wagner, "Experimental Adaptive Beam Forming with Polarization Read-Write Multiplexing using BEAMTAP," The International Topical Meeting on Microwave Photonics MWP 2001, January, 2002.
7. Balakishore Yellampalle, Kelvin Wagner, and Steve Blair, "Anti-guide assisted spatial soliton logic gate," OSA topical meeting on Non-linear Optical Guided Waves, Como, Italy, Sept. 2002.
8. Kelvin Wagner, Friso Schlottau, and Jaap Bregman, "Array imaging using spatial-spectral holography," IOG Information Optics meeting, Mannheim, Germany, Sept. 2002.
9. Alexandre R. S. Romariz and Kelvin Wagner, "Optoelectronic implementation of a FitzHugh-Nagumo neural model," Neural Information Processing System (NIPS), Vancouver, CA, Dec. 2002.
10. Max Colice and Kelvin Wagner, "Phase-cohering holography for coherent analog optical signal processing," GOMAC 03, Tampa FL, Apr. 2003.
11. K. H. Wagner, M. Colice, G. Kriehn, F. Schlottau, and R. T. Weverka, "Photonics Multiple Beam Forming for Broadband RF Antenna Arrays," GOMAC 03, Tampa, FL, Apr. 2003.
12. W. R. Babbit, K. D. Merkel, M. Tian, R. Krishna Mohan, Z. Cole, Y. Sun, Rufus R. L. Cone, K. H. Wagner, and R. W. Equall, "Optical-Coherent-Transient technologies for

wide-bandwidth and multidimensional analog signal processing, GOMAC 03, Tampa, FL, Apr. 2003.

13. Kelvin H. Wagner, Zoya Popovic, Dana Z. Anderson, Randall W, Babbitt, Lloyd Griffiths, Andre Knoesen, "RF Photonic Systems for Array Control and Processing," Invited MURI overview at GOMAC, Monterey CA, March 2002

14. F. Schlottau and K. Wagner, "RF Photonics for Simultaneous Multiple TTD Beamforming for 2-D Antenna Arrays," The International Topical Meeting on Microwave Photonics MWP 2001, January 2002.

15. K. Wagner, Greg Kriehn, Friso Schlottau, Wideband All-optical BEAMTAP, The International Topical Meeting on Microwave Photonics MWP 2001, January 2002.

16. G. Kriehn, F. Schlottau, and K. Wagner, "Optically-Implemented 2-D Beam Steering and Jammer Nulling using BEAMTAP," submitted to ICO Topical Meeting on *Optics in Computing*, SPIE, April 2002.

17. Kelvin H. Wagner, Friso Schlottau, and Jaap Bregman, Array Imaging Using Spatial-Spectral Holography, submitted to ICO Topical Meeting on *Optics in Computing*, SPIE, April 2002.

18. J.A. Hagerty, Z. Popović, Passive Millimeter-Wave Ranging using Discrete Lenses with Wave-Front Coding, *2001 European Microwave Conference Digest,* pp.421-424, London, Sept. 2001.

19. J. Vian, Z. Popović, "Smart lens antenna arrays," 2001 IEEE International Microwave Symposium Digest, pp.129-132, Phoenix, Arizona, 2001.

20. D. Anderson, E. Fotheringham, S. Romisch, P. Smith, Z. Popović, "Smart antenna arrays with adaptive optical processing," IEEE Trans. Antennas and Propagation, Special Issue on Wireless Communications, pp. 607-617, May 2002.

21. P. Smith, E. Fotheringham, D. Anderson, Z. Popović, "Adaptive signal processing for microwave carrier broadband signals," 2003 GOMAC conference digest, Tampa, FL, April 2003.

22. R. Reibel, Z. Barber, M. Tian, W. R. Babbitt, "Temporally Overlapped Linear Frequency Chirp Programming for True Time Delay Applications," poster, Holeburning, Single Molecules, and Related Spectroscopies, November 18-23, 2001, Taipei, Taiwan.

23. M. Tian, R. Reibel, Z. Barber, W. R. Babbitt, "Broadband true-time delay in Tm:YAG," poster, Holeburing, Single Molecules, and Related Spectroscopies, November 18-23, 2001, Taipei, Taiwan.

24. M. Tian, R. Reibel, Wm. R. Babbitt, "Demonstration of broadband true-time delay with optical coherent transient," MWP'2001, January 2002, Long beach, CA.

25. M. Tian, Z. Barber, T. Chang, R. R. Reibel, and W. R. Babbitt* (* Invited Speaker), "The Effects of Optical Nutation on Stimulated Photon Echoes," Holeburning, Single

Molecules, and Related Spectroscopies, November 18-23, 2001, Taipei, Taiwan. (Related work funded by AFOSR)

26. M. Tian*, R. R. Reibel, Z. Barber, J. Fischer, W. R. Babbitt (* Invited speaker), "Optical Coherent Transient True-time delay: Broadband programming methods," 2002 Physics of Quantum Electronics Conference, Snowbird, UT, January 2002.

27. Joe Fischer, Zeb Barber, Randy, Reibel, Mingzhen Tian, and Randy Babbitt, "Linear Phase Chirp Programming for OCT's," Optec2001, (Bozeman, MT, August, 2001)

28. M. Tian, R. R. Reibel, Z. Barber, J. Fischer, and W. R. Babbitt, "Optical Coherent Transient True-Time Delay Generator Programmed with Linear Phase Chirp," in Conference on Lasers and Electro-Optics, Technical Digest (Optical Society of America, Washington DC, 2002), May 2002, Long Beach, CA.

2.6.3   Patents and Applications

1. Invention disclosure submitted: P. Smith, D. Anderson, Z. Popovic, E. Fotheringham, "A dynamic system implementation of analog optical independent component analysis."

2. K. D. Merkel, Z. Cole, K. Rupavaratham, W. R. Babbit and K. H. Wagner, US Patent Application No. PCT/US03/14,612, "Method and Apparatus for Processing High Time-Bandwidth Signals Using a Material with Inhomogeneously Broadened Absorption Spectrum."

2.7   Classes taught by PIs relating to MURI research

1. Dr. Wagner and Dr. Anderson jointly taught a graduate-level Advanced Optics Lab in Fall of 2001 developed for interdisciplinary students with interests in Chemistry, Physics, and Electrical Engineering. Several of the MURI graduate students were involved either as TAs or in taking the class.

2. Zoya Popović taught a class in RF-optical techniques at the Technical University in Munich in 2001. This was a part of an ongoing collaboration. She will teach again in the summer of 2003.

3. Kelvin Wagner taught CU ECEN 6006, Applied Photonics, new course developed to be the second course in the new core optics curriculum. Spring 2002.

4. Kelvin Wagner taught CU ECEN 4016/5016, Fundamentals of Photonics, new introductory course developed as an introduction to the optics core curriculum. This was the second offering of this course, and the first by Prof. Wagner.

5. Randy Babbitt taught MSU Physics 353, Holography-Laser Photography, Spring 2001, undergraduate. Spring 01 (Randy Babbitt), Spring 02 (Randy Reibel taugth it, Randy Babbitt was the mentor).

Optoelectronic Signal Processing Using Finite

Impulse Response Neural Networks

by

Paulo Eduardo H. B. Xavier da Silveira

B.S.E.E., University of Kansas, 1995

M.S.E.E., University of Colorado at Boulder, 1997

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

2001

This thesis entitled:

Optoelectronic Signal Processing Using Finite Impulse Response Neural Networks

written by Paulo Eduardo H. B. Xavier da Silveira

has been approved for the Department of Electrical and Computer Engineering

_____

Kelvin H. Wagner

_____

Timothy X. Brown

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Xavier da Silveira, Paulo Eduardo H. B. (Ph.D., Electrical Engineering)

Optoelectronic Signal Processing Using Finite Impulse Response Neural Networks

Thesis directed by Prof. Kelvin H. Wagner

This thesis investigates the use of finite impulse response neural network as the computational algorithm for efficient optoelectronic signal processing. The study begins with the analysis and development of different suitable algorithms, followed by the optoelectronic design of single-layer and multi-layer architectures, and it is concluded with the presentation of the results of a successful experimental implementation.

First, finite impulse response adaptive filters and neural networks – the algorithmic building blocks – are introduced, followed by a description of finite impulse response neural networks. This introduction is followed by a historical background, describing early optoelectronic implementations of these algorithms. Next, different algorithms capable of temporal back-propagation are derived in detail, including a novel modification to the conventional algorithm, called **delayed-feedback back-propagation**. Based on these algorithms, different optoelectronic processors making use of adaptive volume holograms and three-dimensional optical processing are developed. Two single-layer architectures are presented: **the input delay plane architecture** and **the output delay plane architecture**. By combining them it is possible to implement both forward and backward propagation in two complementary multi-layer architectures: the first making use of the conventional temporal back-propagation and the second making use of delayed feedback back-propagation. Next, emphasis is given to a specific application: the processing of signals from adaptive antenna arrays. This research is initiated by computer simulations of different scenarios with multiple broadband signals and jammers, in planar and circular arrays, studying issues such as the effect of modulator non-linearities to the performance of the array, and the relation between the number of jammers and the final nulling depth. Two sets of simulations are presented: the first set applied to RF antenna arrays and the second set applied to an experimental implementation of a sonar adaptive array.

Experimental results are presented for a single-layer optical processor making use of a scrolling spatial light modulator for representing the input signal and its delayed versions, photorefractive dynamic gratings for implementing the adaptive weights, differential heterodyning for bipolar signal representation, a phase-locked loop for controlling the optical path length, providing long term interferometric stabilization, and acousto-optic modulators for modulating the feedback error signal. Results for multiple beam-forming and jammer nulling are presented for planar and circular adaptive arrays. Finally, it is also shown how one can determine the position of the signal source from the images diffracted from the photorefractive hologram used to store the dynamic weights.

Acknowledgements

To those willing to extend and improve on the ideas here contained, I cordially dedicate this thesis.

The central idea of this thesis, the optoelectronic implementation of finite impulse response neural networks, was inspired by a processor designed by Professor Wagner, published in 1998 [1]. Working on that publication I became acquainted with the temporal back-propagation algorithm, as described by Wan in 1990 [2][1] . I was interested in designing a multi-layer optoelectronic finite impulse response neural network capable of simultaneous forward and backward propagation. This initial effort culminated with the creation of the multi-layer architecture based on the conventional back-propagation algorithm [4], from which the single-layer architectures soon followed. At that time I was already aware of a modification to the LMS algorithm [5], known as the BEAM-TAP method [6]. Following Prof. Wagner's suggestion I used the ideas contained in that modification and extended BEAMTAP to multi-layer finite impulse response neural networks using the same formalism used by Wan, that way creating the delayed feedback back-propagation algorithm.

In September 1999 Dr. Gour Shyam Pati joined our research team as a post-doctoral fellow. From then on we collaborated on several experiments and on a number of publications cited throughout this thesis. His expertise is especially appreciated in recommendations given regarding the experimental use of photorefractive crystals as dynamic volume holograms, and in the operator-notation analysis presented in appendix A.

---

[1] The original idea of back-propagation through time can be traced back to Werbos in 1974 [3].

Contents

Chapter

Figures

Chapter 1

Introduction

The objective of this work is to explore the implementation of optoelectronic systems capable of performing efficient multi-dimensional (spatio-temporal) computation making use of architectures that present significant performance advantages when compared to traditional all-electronic systems. Initially, the design of such systems will be kept general enough so that they may be used in different applications. Next, emphasis will be given to the processing of signals from broadband antenna arrays. Simulations and experimental results will be presented, applied to this specific problem.

Optoelectronic systems combine the capacity for parallel free-space interconnects without frequency dependent cross-talk, characteristics of optics, with the capacity for fast non-linearities at low energy levels, characteristic of electronics. Therefore, successful optoelectronic computing systems must be capable of making use of the advantages presented by each technology. Systems that require many linear operations separated by a layer of fewer non-linear operations are specially attractive, since the linear operations can be easily accomplished in parallel using optics, while electronics can be employed in order to achieve strong non-linearities without becoming a bottleneck to the flow of information. Such an architecture is commonly found in neural networks, where the nodal interconnections usually require a high order of linear operations (N^2 for interconnecting N nodes to N nodes) followed by a lower order of non-linear operations (N), as depicted in figure 1.1.

N inputs

N² interconnects

→ non-linearities

N outputs
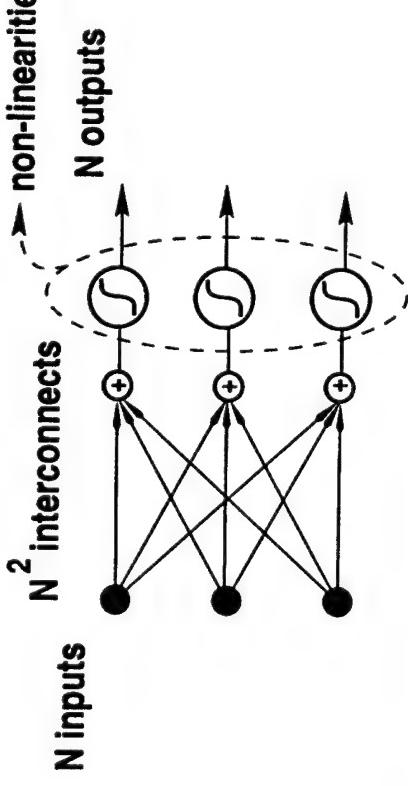
Figure 1.1: Schematic representation of one layer of a conventional neural network in forward propagation. N^2 linear operations are required, which are separated from the N non-linear operations.

When the signal processing task depends on the temporal variation of its input signals, however, traditional neural network processing becomes insufficient, since it considers only the inputs at the present time-step. Finite Impulse Response (FIR) adaptive filters are structures capable of applying time delay to its input in order to perform temporal signal processing. They consist of a single input node which is time-delayed over a tapped-delay line (TDL). Each delayed version of the input is multiplied by a local adaptive weight and the output is given by their sum.

When the interconnects of a neural network are replaced with FIR adaptive filters, we have what is called a **FIR neural network** (FIRNN) [2], extensively discussed in chapter 2. This chapter starts with an algorithmic introduction to FIR filters, neural networks and FIRNNs, followed by a description of optoelectronic devices commonly used in architectures used to implement these algorithms, concluded with an overview of optoelectronic implementations which are representative of previous work related to the research reported on this thesis.

1.1 Gradient descent based algorithms.

This section describes some of the algorithms used to implement adaptive signal processing using FIR adaptive filters, neural networks and FIRNNs, with emphasis given to the gradient descent algorithm and its variations.

Optimization theory [13, 14, 15] provides us with classic tools which can be applied to the search of parameters capable of optimizing some signal processing task, in a procedure which is also known as **adaptive learning** [16]. Of these techniques, special attention is given to the gradient descent method, due to its simplicity of implementation, robustness to noise, and applicability to a wide range of signal processing scenarios.

1.1.1 Cauchy's method

The gradient descent method was first developed by M. Augustin Cauchy in 1847, applied to the generic solution of a system of simultaneous equations [17]. It works as follows: Let $\xi = f(w_1, w_2, w_3, \cdots)$ be a positive and continuous multi-variable function on the parameters $w_1, w_2, w_3, \cdots$. Our goal is to find the parameters which satisfy the equation $\xi = 0$, which must be a minimum, since ξ is positive. The method consists of varying the initial parameters by a small value. That is, let

$$\xi' = f(w_1 + \alpha, w_2 + \beta, \cdots) \cong \xi + \alpha D_{w_1}\xi + \beta D_{w_2}\xi + \cdots, \qquad (1.1)$$

where $D_w\xi$ represents the first derivative of ξ with respect to w, and $\alpha, \beta, \ldots$ are small numbers. Assigning to them the values $\alpha = \eta D_{w_1}\xi, \beta = \eta D_{w_2}\xi, \cdots$, where η is a small positive number, we have

$$f(w_1 - \eta D_{w_1}\xi, w_2 - \eta D_{w_2}\xi, \cdots) = \xi - \eta \left[(D_{w_1}\xi)^2 - (D_{w_2}\xi)^2 + \cdots \right], \qquad (1.2)$$

ξ', the new value of ξ, is guaranteed to be smaller than ξ if η is small enough. As such, if ξ' is not a minimum, the same operation can be applied iteratively until the minimum

is found. In modern notation, this is equivalent to varying the set of parameters $\overline{w} = [w_1\ w_2 \cdots]^T$ to a new value $\overline{w}'$ such that $\overline{w}' = \overline{w} - \eta \nabla_{\overline{w}} \xi$.

When the parameters $w_1, w_2, \ldots$ must satisfy not only one equation, but several ($\xi_1 = 0, \xi_2 = 0, \ldots$), it is sufficient to substitute the resulting system of equations with the single equation $\xi_1^2 + \xi_2^2 + \cdots = 0$, which will have the same properties of ξ defined above (continuous and positive over an interval of interest). Cauchy's method may not provide us with the actual roots of the system of equations. However, it does provide us with a local minimum and, depending on the application, this may be exactly what is sought. In optimization theory, the gradient descent method is applied to problems such as finding the parameters which minimize a cost function or, alternatively, finding the parameters which maximize a performance metric function. As such, the definition of the performance metric can play a fundamental role on the existence of a single or multiple minima, and on the number of iterations necessary to find a set of parameters which yields a solution which is sufficiently close to the minimum.

1.1.2 FIR adaptive filters

In adaptive filtering, an effective performance metric is the mean-square error (MSE), due to its quadratic shape and the consequent existence of a single minimum. This metric is based on the existence of a desired signal $d(\kappa)$ which, when subtracted from the filter output $o(\kappa)$, yields an error $e(\kappa)$ such that the MSE can be defined as

$$\xi(\kappa) = E[e^2(\kappa)] = E[d^2(\kappa) - 2d(\kappa)o(\kappa) + o^2(\kappa)], \tag{1.3}$$

where κ denotes the present integer time step, and $E[e^2(\kappa)]$ denotes the expected value of $e^2(\kappa)$. For a given desired signal $d(\kappa)$, filter adaptation consists of finding an optimum set of parameters $\overline{w}^*$ such that the filter output $o(\kappa)$ most closely resembles $d(\kappa)$.

Figure 1.2 shows a schematic representation of an FIR filter. The filter output is produced from the weighted sum of delayed versions of the input signal $s(\kappa)$, where the
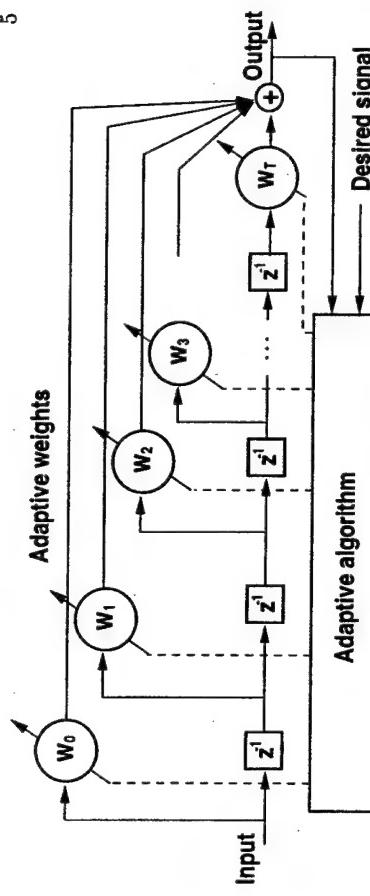
Figure 1.2: Schematic representation of an adaptive FIR filter. The output is given by the weighted sum of delayed versions of the input. Weights are adapted according to an adaptive algorithm and a desired signal. Z^{-1} represents a discrete time delay equal to one time-step.

summation weights $w_\tau(\kappa)$ are the adaptive parameters. That is,

$$o(\kappa) = \sum_{\tau=0}^{T} w_\tau(\kappa)\, s(\kappa - \tau) = \overline{w}^T(\kappa)\, \overline{s}(\kappa), \tag{1.4}$$

where $\overline{s}(\kappa) = [s(\kappa)\ s(\kappa-1)\cdots s(\kappa-T)]^T$, and T is the number of time-delays in the TDL. Note that this operation can also be described by a convolution between the input and the weights, windowed by T.

For a given set of fixed weights, substituting equation 1.4 in equation 1.3, the MSE is given by

$$\xi(\kappa) = E[d^2(\kappa)] - 2\overline{w}^T E[d(\kappa)\,\overline{s}(\kappa)] + \overline{w}^T E[\overline{s}(\kappa)\,\overline{s}^T(\kappa)]\overline{w}$$
$$= d_0 - 2\overline{w}^T \overline{p} + \overline{w}^T \overline{\overline{R}}\,\overline{w}, \tag{1.5}$$

where $d_0 = E[d^2(\kappa)]$ is the power of the desired signal, $p = E[d(\kappa)\,\overline{s}(\kappa)]$ is the cross-correlation vector between the desired and the input signals, and $\overline{\overline{R}} = E[\overline{s}(\kappa)\,\overline{s}^T(\kappa)]$ is the correlation matrix of the input signal. Note that ξ is a quadratic function of $\overline{w}$, which allows for a simple solution when $\overline{p}$ and $\overline{\overline{R}}$ are known. The gradient of the MSE

with respect to the parameters $\overline{w}$ is given by

$$\nabla_{\overline{w}}\xi = \left[\frac{\partial \xi}{\partial w_0} \; \frac{\partial \xi}{\partial w_1} \; \dots \; \frac{\partial \xi}{\partial w_T}\right]$$
$$= -2\overline{p} + 2\overline{\overline{R}}\,\overline{w}. \qquad (1.6)$$

Solving for $\nabla_{\overline{w}}\xi = 0$ and assuming $\overline{\overline{R}}$ to be non-singular, the optimum set of weights is given by

$$\overline{w}^o = \overline{\overline{R}}^{-1}\,\overline{p}. \qquad (1.7)$$

This is known as the Wiener solution and $\overline{w}^o$ is the optimum set of weights. However, in practice it is very difficult and time consuming to acquire a precise estimate of $\overline{p}$ and $\overline{\overline{R}}$. Instead, gradient descent adaptation can be performed, such that the weights are adapted according to the rule

$$\overline{w}(\kappa+1) = \overline{w}(\kappa) - \eta\nabla_{\overline{w}}\xi$$
$$= \overline{w}(\kappa) + 2\eta\left[\hat{\overline{p}}(\kappa) + \hat{\overline{\overline{R}}}(\kappa)\overline{w}(\kappa)\right], \qquad (1.8)$$

where $\hat{\overline{p}}(\kappa)$ and $\hat{\overline{\overline{R}}}(\kappa)$ are current estimates of the cross-correlation vector and the autocorrelation matrix. For on-line gradient descent, let these estimates be simply given by their instantaneous values. That is, $\hat{\overline{p}}(\kappa) = d(\kappa)\overline{s}(\kappa)$ and $\hat{\overline{\overline{R}}}(\kappa) = \overline{s}(\kappa)\overline{s}^T(\kappa)$. Then, the gradient estimate is simply given by $\nabla_{\overline{w}}\xi = -2e(\kappa)\overline{s}(\kappa)$, which is exactly the same gradient we would obtain if the metric function would be substituted by the instantaneous squared error $e^2(\kappa)$ instead of the MSE. Consequently, this method of gradient descent is also known as the **least-mean-square** (LMS) algorithm [5, 18], which is very popular due to is small computational complexity (an order of T operations are required at each time step) and robustness. Substituting $\nabla_{\overline{w}}\xi$ in equation 1.8, we come up with the LMS weight update rule:

$$w_\tau(\kappa+1) = w_\tau(\kappa) + \eta e(\kappa)s(\kappa - \tau), \qquad (1.9)$$

where the constant value of 2 has been included into η.

FIR filters have been successfully used in the solution of problems such as signal enhancement [19], channel equalization [20], system identification [21], linear prediction [22] and channel modeling [23].

1.1.3 Neural networks

Neural networks have the remarkable capability for processing high-dimensional data with a fairly limited number of operations, while simultaneously being capable of interpolating a large amount of data from a limited training set. They have been successfully used in numerous applications, such as pattern recognition [16], pattern classification [24, 25], function approximation [26, 27], control systems [28], and associative memories [29, 30], to name a few. Traditionally, they are composed of several interconnected layers. In each layer the input nodes are connected to the output nodes through a weighted sum, where the weights are the adaptive parameters of the network, and a nonlinear activation function $f(\cdot)$ (usually a sigmoid, of the form $f(x) = 1/[1 + \exp(-x)]$, such that the for classification problems [31]) is applied to each output node (Fig. 1.3), such that the output $o(\kappa)$ for a 2-layer network is given by

$$o_m(\kappa) = f\left[\sum_{j=1}^{N^{(1)}} w_{mj}^{(1)}(\kappa)\, f\left[\sum_{i=1}^{N^{(0)}} w_{ji}^{(0)}(\kappa)\, s_i^{(0)}(\kappa)\right]\right], \qquad (1.10)$$

where $w_{ji}^{(l)}$ denotes the adaptive weight interconnecting node i to node j at layer l. In vector form, it can be written as

$$\overline{o}(\kappa) = f\left[\overline{w}^{(1)}(\kappa)\, f\left[\overline{w}^{(0)}(\kappa)\, \overline{s}^{(0)}(\kappa)\right]\right]. \qquad (1.11)$$

The calculation of the final output as the input is propagated through layers is called **forward propagation**, and it requires $\sum_l N^{(l)} N^{(l+1)}$ operations, where $N^{(l)}$ is the number of nodes present at layer l. The operation of modifying the interconnection weights in order to cause the network to perform a desired function is called **learning**. In **supervised learning** sets of previously known input-output pairs are used

where

$$\sigma_j = \sum_{i=1}^{N} w_{ji} s_i,$$ (1.13)

is the pre-synaptic linear output of layer 0 before undergoing the objective function $f(\cdot)$. The second term in the right side of equation 1.12 can be calculated by using equation 1.13, giving us $\frac{\partial \sigma_j}{\partial w_{ji}} = s_i$. Let us define the first term as $\frac{\partial E^n}{\partial \sigma_j} \equiv \delta_j$, then let us determine δ_j. In the output layer it is given by

$$\delta_m = \frac{\partial E^n}{\partial \sigma_m} = f'(\sigma_m) \frac{\partial E^n}{\partial o_m},$$ (1.14)

where $f'(\sigma_m)$ is the first derivative of $f(\sigma_m)$ with respect to σ_m. For hidden units, we make use of the chain rule of derivatives and obtain

$$\delta_j^{(l)} = \frac{\partial E^n}{\partial \sigma_j^{(l)}} = \sum_{m=1}^{N^{(l+1)}} \frac{\partial E^n}{\partial \sigma_m^{(l+1)}} \frac{\partial \sigma_m^{(l+1)}}{\partial \sigma_j^{(l)}} = f'(\sigma_j^{(l)}) \sum_{m=1}^{N^{(l+1)}} w_{mj}^{(l+1)} \delta_m^{(l+1)},$$ (1.15)

which tells us that the δ term for a hidden unit can be obtained simply by propagating the δ term from the next layer back through the weights which interconnect the two layers. Finally, the weights are adapted following the gradient descent rule, i.e.

$$w_{ji}^{(l)}(\kappa+1) = w_{ji}^{(l)}(\kappa) - \eta \delta_j^{(l)}(\kappa) s_i^{(l)}(\kappa)$$ (1.16)

Note that back-propagation is an operation which, as forward propagation, also requires $\sum_l N^{(l)} N^{(l+1)}$ computations. This is far less than the $\sum_l (N^{(l)} N^{(l+1)})^2$ steps required by other algorithms, such as weight perturbation, which require a forward propagation step after a small modification of each weight.

1.1.4 The finite impulse response neural network

Finite impulse response neural networks, also known as time-delay neural networks, can be described as a set of neural networks that make use of FIR adaptive filters as intra-layer interconnects, thus being capable of simultaneous spatial and temporal adaptation. Besides all the applications usually assigned to FIR adaptive filters and

---

Figure 1.3: Schematic representation of a two-layer neural network in forward propagation. At each layer, the input signal vector undergoes a weighted sum followed by a non-linear operation, where the weights are the adaptive parameters.

to modify the weights according to the rules set by the learning algorithm. In **on-line learning** weight updates take place simultaneously with forward propagation. Different paradigms exist for weight adaptation, but a common one is the same gradient descent used in FIRs [16]. In neural networks, because of the non-linear operation in forward propagation, it is usually the case that it is not possible to find a closed form solution to the optimum parameters, such as the Wiener solution found for FIRs. Therefore, the use of iterative algorithms becomes a necessity.

A computationally efficient form for calculating the gradient of the weights in the inner layers with respect to the output in the final layer is the so called **back-propagation** algorithm, first developed by Werbos [3] and independently developed by Rumelhart et al [32]. In brief, it works as follows: first we define an error metric (the function we want to minimize) $E = \sum_n E^n$, where E^n is a differentiable function of the output $\bar{o}$ when the nth training pattern is present at the input. The gradient of the error function with respect to the adaptive weights is then given by

$$\frac{\partial E^n}{\partial w_{ji}} = \frac{\partial E^n}{\partial \sigma_j} \frac{\partial \sigma_j}{\partial w_{ji}},$$ (1.12)

Figure 1.4: Schematic representation of a layer of a FIRNN in forward propagation. N^2T adaptive weights are necessary to interconnect the input nodes to the output nodes.

neural networks, FIRNNs also find applications in real-time speech recognition [33, 34], multi-dimensional time-series prediction [35, 36], state machine implementation [37], control systems [38], function estimation [39], inverse filtering equalization [40], and true-time delay adaptive antenna arrays [41]. In fact, any application which requires the simultaneous processing of spatially and temporally varying data is a potential application for FIRNNs. Their added ability, however, directly translates into higher computational complexity due to the higher dimensionality of the spatio-temporal input signals and adaptive weights. However, this thesis shows that such a higher complexity can be effectively mapped into three-dimensional optical processing using volume holograms, and mapped into optical architectures which are far more computationally efficient than those used for implementing conventional neural networks.

Algorithms for FIRNNs that make use of gradient descent and the efficient back-propagation algorithms are of special interest, and they are studied in detail in chapter 2. For now, note that in forward propagation, the output of FIRNNs will be given by the

summation of the outputs of individual FIRs, i.e.

$$o_j(\kappa) = f\left[\sum_{i=1}^{N}\sum_{\tau=0}^{T} w_{ji,\tau} s_i(\kappa-\tau)\right].$$ (1.17)

As such, they can be described as the summation of multiple convolutions between the input signal and the adaptive weights. Analogously, efficient back-propagation can be described by the correlation between the error signals and the same adaptive weights.

1.1.5 Adaptive antenna arrays

Adaptive antenna array processors consist of systems capable of adaptively combining the signals from multiple antenna elements in order to implement a required task, which usually consist of the optimum detection of desired signals constrained by the rejection of undesired signals (i.e. noise sources, intentional jammers, neighboring interference, multi-path interference, etc) [42]. They can be described as a single-layer, linear FIRNN, usually with a single output [41]. As such, they provide us with an application for FIRNNs which is simple enough for experimental implementation in a reasonable amount of time, and yet renders useful results.

Figure 1.5 shows a diagram of a time-delay-and-sum adaptive array. Each input signal undergoes an FIR filter and their combined outputs is summed, forming the final output $o(\kappa)$ given by

$$o(\kappa) = \sum_{i=1}^{N}\sum_{\tau=0}^{T} w_{i,\tau} s_i(\kappa-\tau),$$ (1.18)

where N is the number of antenna elements in the array and T is the total number of time delays. Real-time adaptive processing of wide-band signals in large antenna arrays is a demanding signal processing task, since wide-band signals require the use of a large number of time-delays in order to avoid beam-squint (the variation of the AOA of a broadband signal with frequency when detected by a phased antenna array without true-time-delay). This task becomes especially difficult when multiple broadband signals are present — requiring more degrees of freedom and, therefore, more processing capacity

Complex Adaptive Weights
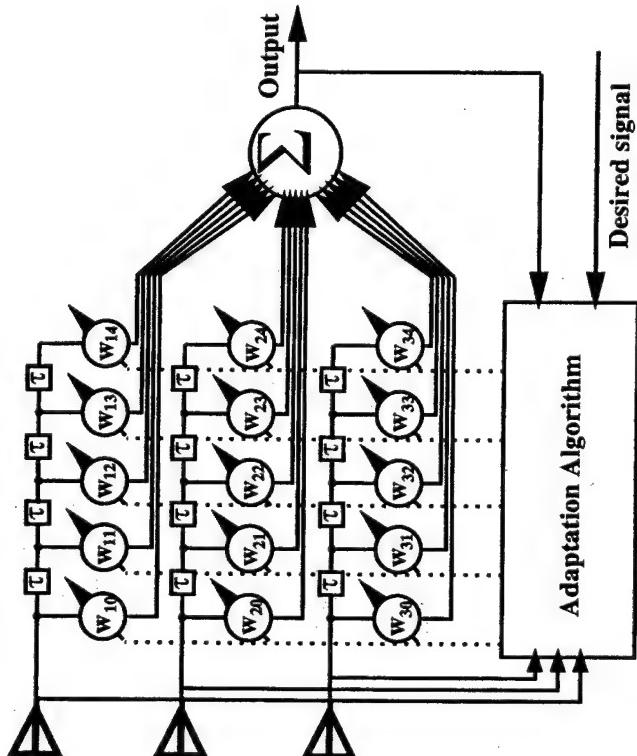
Figure 1.5: Adaptive array processor. Each input undergoes an FIR filter, where the summation of their outputs produces the final output. The weights are controlled by an adaptive algorithm.

— in the near-field of the array — preventing the use of simple far-field beam-forming algorithms, or when the array shape is time varying.

In order to provide the array with adaptability to the signal environment, the FIR weights are controlled by an adaptive algorithm, such as LMS gradient descent (Fig. 1.6). Such an array processor was first proposed by Widrow et al [5] in 1967. It consists of an extension of the basic LMS algorithm used for FIRs in order to allow for simultaneous spatial and frequency (temporal) filtering. It works as follows: we first define the signal input as a rastered vector containing all the delayed inputs from all

Figure 1.6: Adaptive array processor controlled by the LMS algorithm. The weights accumulate the product between the local input and the error signal.

antenna elements, i.e.

$$\overline{s}(\kappa) = \left[\begin{array}{cccccccc} s_1(\kappa) & s_1(\kappa-1) & \cdots & s_1(\kappa-T) & s_2(\kappa) & \cdots & s_2(\kappa-T) & \cdots & s_N(\kappa-T) \end{array} \right]^T . \quad (1.19)$$

Then we define a weight vector $\overline{w}$ containing the rastered representation of all the weights,

$$\overline{w} = \left[\begin{array}{cccccccc} w_{10} & w_{11} & \cdots & w_{1T} & w_{20} & \cdots & w_{2T} & \cdots & w_{NT} \end{array} \right]^T . \quad (1.20)$$

The output can now be represented by $o(\kappa) = \overline{w}^T \overline{s}(\kappa)$, and we are back to the case of equation 1.4. The LMS learning rule can be applied, providing us with the following

weight update equation.

$$\bar{w}(\kappa+1) = \bar{w}(\kappa+1) + 2\eta e(\kappa)\bar{s}(\kappa),$$ (1.21)

which, in continuous time, can also be written as

$$\bar{w}(t) = 2\eta \int_0^t e(t')\bar{s}(t')\,dt'.$$ (1.22)

That is, a time integration of the product between the error and the input signal, as indicated by the inset in figure 1.6. We can get further insight into equation 1.22 by writing it in unrastered index notation, i.e.,

$$w_{i\tau}(t) = 2\eta \int_0^t e(t')s_i(t'-\tau\Delta_t)\,dt'.$$ (1.23)

We now realize that the weights are given by a correlation between the input and and error signal, where Δ_t is the time interval between time-steps. Substituting into equation 1.19, we get

$$o(\kappa) = \sum_{i=1}^N \sum_{\tau=0}^T s_i(t-\tau) \int_0^t s_i(t'-\tau\Delta_t)\,[d(t')-o(t')]\,dt'.$$ (1.24)

The operation described by equation 1.24 is known as a **correlation-cancellation loop** (CCL). That is, feeding back the error signal during weight adaptation has the effect of correlating the desired signal with the input while simultaneously canceling (filtering out) uncorrelated signals. Such an operation is useful when trying to detect a desired signal while nulling out interference sources, such as intentional jammers.

Adaptive antenna arrays are treated in more detail in chapter 4, where different input scenarios and implementation issues are discussed and simulation results are presented.

1.2 Optoelectronic components

From the analysis presented in section 1.1, we realize that the experimental implementation of FIRNNs requires the use of optoelectronic components capable of pro-

viding us with time delay and a variable representation of the adaptive weights. Furthermore, we note that the forward and backward propagation operations require the implementation of correlations and convolutions. Thus, time-integrating (TI) [43, 44] and space-integrating (SI) [45] architectures are of special interest due to their capacity for performing fast correlations and convolutions in the optical domain, thus taking advantage of the capacity for linear, high-dimensional signal processing offered by optical implementations. Chapter 3 describes in more detail the optical architectures used to implement FIRNNs. For now, let us have a brief description of the optoelectronic components commonly used in the implementation of the adaptive algorithms described in section 1.1.

1.2.1 Acousto-optic modulators

Acousto optic modulators (AOM) operate by launching an acoustic wave along a medium which exhibits the photoelastic effect. The traveling acoustic wave induces a strain in the medium, which causes a change in its index of refraction n given by [46]

$$\Delta\eta_{ij} = \Delta\left(\frac{1}{n^2}\right)_{ij} = p_{ijkl}S_{kl},$$ (1.25)

where $\Delta\eta_{ij}$ is the change in the optical impermeability tensor, S_{kl} is the strain tensor resulting from the traveling acoustic wave, and p_{ijkl} is the strain-optic tensor. Since S_{kl} is usually small (typically in the order of 10^{-5}), additional nonlinear terms of equation 1.25 are usually safely ignored.

When light propagates through the same medium, the variation of the index of refraction causes a modulation of its phase. When the modulation is periodic, light is diffracted at an angle proportional to the modulation frequency, and with an amplitude proportional to the amplitude of that modulation. For a sufficiently thick interaction region, Bragg matched diffraction occurs, concentrating most of the diffracted light into
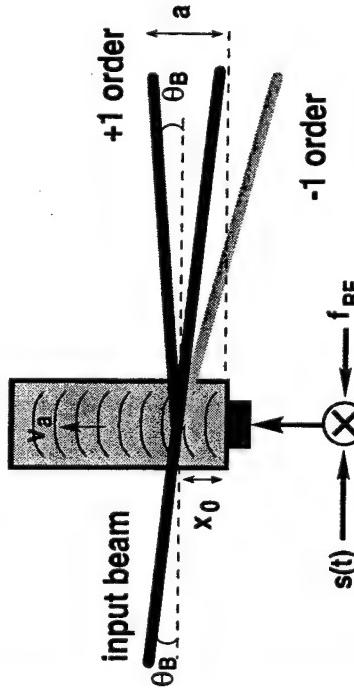
Figure 1.7: An acousto-optic modulator diffracting light at the Bragg angle. The complex field amplitude of the diffracted light is proportional to $s(t)$.

a single order at the angle θ_B given by

$$\theta_B = \arcsin\left(\frac{\lambda}{2n\Lambda}\right),$$ (1.26)

where λ is the wavelength of the incident light beam and $\Lambda = \frac{v_a}{f_{RF}}$ is the wavelength of the acoustic wave, where v_a is the velocity of the acoustic wave in the modulating medium and f_{RF} is the RF center frequency of the modulator.

Figure 1.7 describes the operation of an AOM. Most of the input beam is undiffracted, but the strongest diffraction is present at the Bragg angle with a diffraction efficiency η_a. The AOM is fed with a signal of interest $s(t)$, mixed at f_{RF}, the frequency at which the modulator has its highest efficiency. Assuming linear operation, and that the highest frequency component of $s(t)$ is much smaller than f_{RF}, the complex field amplitude of the light diffracted at the Bragg angle, at the +1 order, is given by

$$E_s(\bar{r},t) = \sqrt{\eta_a}\,g\left(\frac{x_0}{w_0}\right)\frac{E_0}{2}e^{-2\pi j[\nu t-(\bar{k}_i+\bar{k}_s)\cdot\bar{r}]}\left[s\left(t-\frac{x_0}{v_a}\right)\right]e^{-2\pi j f_{RF}t} + c.c.,$$ (1.27)

where ν is the optical frequency, $g(x) = \exp\{-x^2\}$ represents the Gaussian profile of the illuminating beam, and w_0 represents its $1/e$ width. $\bar{k}_i = \frac{2\pi\nu n}{c}(-\hat{x}\sin\theta_B + \hat{z}\cos\theta_B)$ is

the wave-vector of the incident field, and $\bar{k}_g = \frac{f_{RF}}{v_a}\hat{x} = \frac{2\nu n\sin\theta_B}{c}\hat{x}$ is the acoustic grating vector. Expanding the term $(\bar{k}_i + \bar{k}_g)\cdot\bar{r}$, we get

$$E_s(\bar{r},t) = \sqrt{\eta_a}\,g\left(\frac{x_0}{w_0}\right)s\left(t-\frac{x_0}{v_a}\right)\frac{E_0}{2}e^{-2\pi j[(\nu+f_{RF})t-\frac{\nu n}{c}(z\cos\theta_B+x\sin\theta_B)]} + c.c.,$$ (1.28)

That is, a field propagating in the $+\theta_B$ direction. Note that the output signal is delayed by x_0/v_a, allowing AOMs to be used to generate variable RF time-delays. Also note that the output field is frequency shifted by f_{RF} with respect to the input field $E_i(\bar{r},t) = E_0\cos(2\pi\nu t - \bar{k}_i\cdot\bar{r})$. This is due to the Doppler shift caused by diffraction from a moving acoustic wavefront, and it must be taken in consideration when the modulated beam is mixed with another beam in a time-integrating architecture, as described in chapter 5.

1.2.2 Photorefractive crystals

The photorefractive (PR) effect occurs in electrooptic photoconductive crystals and results in a change in the index of refraction of a material that is proportional to the gradient of the intensity pattern of interfering beams [7, 47, 48, 49, 50]. It takes effect in impurity-doped electrooptic crystals, as described in figure 1.8. When two mutually coherent beams interfere in such a crystal, the consecutive constructive and destructive interference produces a sinusoidal spatial variation of the intensity of light throughout the volume of the crystal (Fig. 1.8(a)). The optical intensity causes charge carriers to be excited from occupied donor states. These free carriers then drift or diffuse and are retrapped at an empty donor, causing the regions illuminated with high intensity to lose carriers, while the regions illuminated with low intensity acquire an excess of trapped electrons. This results in a space charge distribution ρ_{sc} (Fig. 1.8(b)) which, according to Poisson's equation $(\nabla\cdot\bar{E}_{sc} = \frac{\rho_{sc}}{\epsilon_0})$, results in an electric field $\bar{E}_{sc}$ shifted by $\pi/4$ with respect to ρ_{sc} [Fig. 1.8(c)]. The electrooptic effect induces the formation of a grating $\Delta n \propto rE_{sc}$, where r is the appropriate electrooptic coefficient.

The grating $\Delta n(x,y,z,t)$ that is formed within the crystal results in two beam

where $\nabla_\perp^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ and $\bar{I}_{dc} = \sum_i |E_i|^2 + I_d$, $\quad i,j=1,2,\quad i\neq j$

where $E_1(x,y,z,t)$, and $E_2(x,y,z,t)$ are the space and time-varying amplitude distributions of the interfering beams. β represents the sensitivity of the PR crystal (i.e. the change in index of refraction per unit energy per unit area), and $\tau = \bar{I}\tau_{pr}$ is the intensity normalized time response, with τ_{pr} denoting the PR time constant. I_d is the normalized dark intensity that accounts for thermal excitation of carriers and/or incoherent erasure. The space-time evolution of the refractive index modulation in the PR medium can be obtained from the solution of equation 1.29, and is given by

$$\Delta n(x,y,z;t) = \beta \int_0^t E_1(x,y,z;t') E_2^*(x,y,z;t') e^{-\frac{(t-t')}{\tau_{pr}}}\, dt'. \quad (1.31)$$

The integral in the above equation represents the finite time-integrated response of the PR crystal with a linear superposition of all transient gratings scaled by an exponential decay factor $\exp(-\frac{t-t'}{\tau_{pr}})$. Such a response also indicates the low-frequency characteristic of the PR medium, i.e., if the frequencies of the time varying amplitude distributions in the writing beams differ from each other by $1/\tau_{pr}$, it will not result in a significant index modulation. As such, it is possible to use PR crystals as time-integrating correlators [52].

The photorefractive volumetric grating can also be used to represent the interconnections between the interfering beams used to record it [53, 54], such that a beam reading out the resulting grating will be diffracted in the direction of the other beam used to record it. From equation 1.31 we see that it is possible to dynamically modify the PR grating and, by modifying signals in the recording beams, it is possible to use PR crystals to implement adaptive interconnects. The optical momentum surface (also known as $\bar{k}$-space) of a uniaxial crystal, describing the allowed momenta of propagation of an optic beam of angular frequency ω is given by [46]

$$\left(\frac{k^2}{n_o^2} - \frac{\omega^2}{c^2}\right)\left(\frac{k_x^2+k_y^2}{n_e^2} + \frac{k_z^2}{n_o^2} - \frac{\omega^2}{c^2}\right) = 0, \quad (1.32)$$

where n_o and n_e are the ordinary and extraordinary indexes of refraction of the medium, respectively. Expression 1.32 describes a double sheeted surface of propagating wave

---

Figure 1.8: The photorefractive mechanism (from [7]). (a) Intensity interference pattern. (b) Space charge distribution at steady state. (c) Spatial distribution of the change in the refractive index.

coupling of the interfering beams, such that the dynamics of the index grating modulation in the PR crystal can be described by the following set of coupled differential equations [51]

$$\frac{\partial \Delta n(x,y,z;t)}{\partial t} + \frac{\bar{I}_{dc}}{\tau}\Delta n = \beta E_1 E_2^* \Delta n \quad (1.29)$$

$$\left(\frac{\partial}{\partial z} - \frac{i}{2k}\nabla_\perp^2\right) E_i = -i\Delta n E_j, \quad (1.30)$$

Figure 1.9: The optical momentum surface. The input plane waves in the directions of $\bar{k}_i$ and $\bar{k}_j$ interfere, forming the grating $\overline{K}_{ji}$. Bragg matching dictates that the holographic readout wave must be in the direction of $\bar{k}_j$ in order to produce a diffracted beam in the direction of $\bar{k}_i$. Bragg degeneracy allows for the readout beam to be rotated about $\overline{K}_{ji}$, which results in a strong diffraction rotated similarly about $\overline{K}_{ji}$.

vectors. The first factor gives us the wave vectors for ordinary waves, which describes a sphere of radius $k = \omega n_o/c$. The second factor gives us the wave vectors for extraordinary waves, which describes an ellipsoid of axes $k_z = \omega n_o/c$ (defined as the crystal axis $\bar{c}$), and $k_x = k_y = \omega n_e/c$ (Fig. 1.9).

Let $\bar{k}_i$ and $\bar{k}_j$ be the wave vectors of two plane waves interfering and producing a first order index grating proportional to $\cos(\overline{K}_{ji}.\bar{r}+\phi)$, where ϕ is the phase shift between the grating and the interference pattern between the incident waves, and $\overline{K}_{ji} = \bar{k}_j - \bar{k}_i$ is the grating vector. Different points in the momentum surface describe solutions for propagating waves at different directions. Thus, for a monochromatic wave to propagate, its wave vector must have its tip "touching" a point in the momentum surface. This

means that if the grating is readout by a plane wave propagating in the same direction as $\bar{k}_i$, the diffracted wave will propagate in the direction of the wave vector $\bar{k}_i + \overline{K}_{ji} = \bar{k}_j$, since this is the only possible solution (the other possibility, $\bar{k}_i - \overline{K}_{ji}$, renders a vector outside the momentum surfaces). Thus, $\bar{k}_j = \overline{K}_{ji} + \bar{k}_i$ is known as the Bragg condition for holographic readout.

Analogously, if the vector $\bar{k}_j$ is used for readout, the diffracted wave will be in the direction $\bar{k}_j - \overline{K}_{ji} = \bar{k}_i$. Thus, the holographic recording of a volume grating between wave vectors $\bar{k}_j$ and $\bar{k}_i$ created the means to spatially interconnect an optical beam propagating in the direction of $\bar{k}_i$ towards the direction of $\bar{k}_j$, and vice-versa. But what happens if a different wave vector is used for readout? A diffracted wave is only efficiently produced if the Bragg condition is satisfied. However, infinite pairs of degenerate input-output wave vectors exist that are matched by the same grating vector located around the so called degeneracy ellipses. In the example shown in figure 1.9, if the wave vectors $\bar{k}_i$ or $\bar{k}_j$ are both simultaneously rotated about the grating $\overline{K}_{ji}$, the Bragg condition would still hold and their tips would still be located at points in the momentum surface. Thus, when volume holograms are used as weighted interconnects care must be taken in order to prevent or to account for the effects due to Bragg degenerate readout. This can be done by using fractal sampling grids, which uniquely address individual uncertainty boxes in K-space [53, 55], or by using geometries in which degenerate diffractions are either absent or are present in positions that do not affect other interconnections, as done in the architectures presented in chapter 3.

1.2.3 Spatial light modulators

Spatial light modulators (SLM) are devices capable of modulating the amplitude or phase of a beam of light which is either transmitted through or reflected by it. The modulation varies at different spatial positions and it may also vary in time. Of special interest to us are the ferroelectric liquid crystal SLMs (FLC-SLM), due to their combined

temporal integration, thus offering us an alternative device for implementing TI architectures.

1.3 Previous optical implementations

After a brief introduction to algorithms and optoelectronic components relevant to the optoelectronic implementation of FIRNNs, we are now capable of reviewing some of the previous work that is related to the work reported on this thesis. Let us start by describing experimental implementations of FIR adaptive filters, followed by some of the relevant implementations of neural networks. Then a unique FIRNN implementation will be described, concluding with some remarks regarding the optoelectronic implementations of adaptive antenna arrays.

1.3.1 Optoelectronic implementations of FIR adaptive filters

Adaptive FIRs have been optically implemented using a combination of time-integrating and space-integrating architectures [57, 8, 58, 9]. Figure 1.10 depicts the architecture developed by Rhodes [8], which implements an adaptive linear predictor making use of a correlation-cancellation loop. An electro-optic (EO) modulator is used to modulate the incident laser light with the error signal, which is followed by a Bragg cell (BC1), which modulates the incident beam with delayed versions of the input signal $s(t)$. The liquid crystal light valve temporally accumulates the incident field, performing a temporal integration which provides the required weights, described by

$$w_z(t) = \eta \int_0^t e(t')\, s(t' - x/v_a) \exp\left(\frac{t'-t}{\tau}\right)\, dt', \tag{1.33}$$

where τ is the time constant of the light valve.

The second part of the processor makes use of spatial integration in order to calculate the output signal. A vector-vector multiplier is implemented by reading out the weights from the LCLV with a reference beam. The LCLV modulates the polarization

characteristics of high spatial resolution, high contrast ratio, electronic addressing (thus working as an electro-optical transducer) and fairly high operating speed of a few kHz, if compared to conventional nematic liquid crystal SLMs, which operate at only up to 100 Hz. Their operation is described in more detail in chapter 5.

Another device of interest is the liquid crystal light valve (LCLV). It is an optically addressed device, that is, illumination in the "write" side of the device affects the retardation at the "read" side of the device, thus producing a corresponding spatial modulation in a beam reflected by the read side. Similar to photorefractives, their slow time response provides them with the ability to generate a modulation pattern which is proportional to the temporal integration of the signals detected on the write side. As such, it is possible to use them to implement the adaptive interconnection between the input and output planes of the reflected beam.

1.2.4 Time-delay and integrate photodetectors

Time-delay-and-integrate (TDI) photodetectors are devices capable of detecting light by generating charge carriers, spatially shifting the detected carriers, and accumulating the shifted carriers with new ones generated from the current detection. They can be implemented by a charge-coupled-device (TDI-CCD), or by a traveling-fringe-detector (TFD) [56]. In a TDI-CCD, the CCD shift-register scrolls only one row at a time and only transfers the detected charges to the output differential amplifier at the last row, instead of transferring all the charges at every frame, as is commonly done in a conventional CCD detector. They are capable of a large number of individual shift positions (2^{11} or more), but are conventionally limited to a maximum operating frequency of a few MHz. In a TFD the charge carriers are accelerated by a difference in potential between the connecting electrodes, continuously picking up new carriers as they move, implementing a lossy, but fast (in the GHz range) TDI photodetector.

Note that a TDI detector provides us not only with time delay, but also with

Figure 1.11: Optoelectronic adaptive filter in which the adaptive weights are implemented by time integration in a photorefractive crystal, and space integration is used for output calculation (from [9]). BS – beam-splitter, BC – Bragg cell, PR Xtal – photorefractive crystal.

integrated at the output detector, producing the signal $o(t)$, given by

$$o(t) = \int G(x,t)\, s^*(t - x/v_a)\, dx + c.c.. \qquad (1.35)$$

The "write" laser is set at a different wavelength from the "read" laser, such that an optical filter provides the necessary read-write isolation. The beam from the "write" laser is modulated by the error signal $e(t)$, which is then modulated by the input signal at the BC. The PR crystal temporally integrates the resulting product, accumulating the following grating

$$G(x,t) = \beta \int_0^t e(t')\, s(t' - x/v_a) e^{\frac{t'-t}{\tau_{pr}}}\, dt', \qquad (1.36)$$

where β represents the PR sensitivity and τ_{pr} is the PR time constant. This way, the PR grating accumulation performs the necessary correlation cancellation operation.

1.3.2 Optoelectronic implementations of Neural networks

Several optical implementations of neural networks have been proposed in the literature [59, 10, 11, 60, 61]. Figure 1.12 depicts an implementation of a two-layer neural

Figure 1.10: Optoelectronic adaptive filter where the adaptive weights are implemented by time integration in a LCLV (from [8]). Space integration is used in order to calculate the output. EO – electro-optical modulator, SF – spatial filter, C – cylindrical lens, BC – Bragg cell, LCLV – liquid crystal light valve, POL – polarizer, PMT – photomultiplier tube.

of the read-out beam and a polarizer (POL) converts the polarization modulation into amplitude modulation. The LCLV is imaged onto BC2 which modulates the read-out beam with delayed versions of the input signal. The output is then spatially integrated in the photomultiplier tube (PMT), providing an output signal proportional to

$$o(t) = \int w_x(t)\, s(t - x/v_a)\, dx. \qquad (1.34)$$

The output signal is electronically subtracted from the desired signal $d(t)$, providing the feedback error signal $e(t)$ used in the weight accumulation, closing the correlation cancellation loop.

Figure 1.11 depicts the architecture developed by Montgomery and Lange, used to implement an adaptive narrow-band interference canceler, also making use of the LMS algorithm. Light from the "read" laser is modulated with delayed versions of the input signal at the BC. The modulated light is diffracted by a grating $G(x,t)$ in the PR crystal, which represents the adaptive weights. The diffracted light is then spatially

Figure 1.12: A two-layer neural network, implemented by Toyoda *et al* [10]. The first layer performs feature extraction, while the second layer performs a pattern recognition task. CRT – cathode ray tube, PTR – phototransistor, LED – light emitting diode, MSLM – micro-channel spatial light modulator.

network used in pattern recognition, as described by Toyoda *et al*[10]. In the first layer, a TV camera captures the input image and displays it in a cathode ray tube (CRT), which modulates an optically addressable micro-channel spatial light modulator (MSLM) [62], performing an incoherent to coherent representation of the input image. The resulting coherent image is then optically Fourier transformed, and a phototransistor (PTR) array detects a small 4×4 area of the resulting transform.

The second layer consists of an optical associative memory in which MSLM1 stores the expanded image of the weight matrix W' and MSLM2 stores the input image captured from the first layer, repeated 4 times in a 16×16 element array. Forward propagation is performed optically, where the weight matrix is multiplied by the input matrix, producing all inner products between the two matrix elements. These products are then added in an electrical/parallel circuit, producing the necessary matrix multiplication between the input and weight matrices. Learning is performed in a computer, where the output Y is compared with a target input $\bar{Y}$, producing an error term. The

weight matrix is then updated according to the rule:

$$W'(\kappa+1) = W'(\kappa) + \eta(\bar{Y}_{mag} - Y_{mag}) \otimes S_{mlt}, \qquad (1.37)$$

where S_{mlt} denotes the input matrix repeated 4 times, Y_{mag} denotes the output vector magnified by a factor of 4, and $\otimes$ denotes the point by point multiplication of the spatially expanded matrices followed by electronic summation of the subarrays, resulting in an operation functionally equivalent to conventional matrix multiplication. Note that $\bar{Y}_{mag} - Y_{mag} = e(\kappa)$ and, therefore, this learning rule is simply another way of stating the LMS rule.

Figure 1.13 depicts a 2-layer optical neural network implemented by Psaltis and Qiao [11]. During training of the first layer, shutter S1 is closed and random images are displayed in a monitor and are made coherent by LCLV1. The resulting coherent image is split in two by beam-splitter (BS)1 and the two images interfere at PRC1, which is a high gain crystal prone to a large amount of beam fanning [63]. As a result, a grating is formed which performs a dimensionality expanding random mapping. After completion of the training of the first layer, the resulting grating is thermally fixed [64].

The second layer is trained in order to map distinct input patterns to distinct spatial positions in the output CCD. This is done by displaying each one of the training patterns at the input monitor and LCLV1 while shutter S3 is closed. This pattern is diffracted by the PRC1, producing a pattern of higher dimensionality at the LCLV2. A plane wave reads out the image at the LCLV2, completing forward propagation through the first layer. The resulting pattern then records a grating in the PRC2 along with a plane wave reference beam which is focused onto the CCD. The tilt of the reference beam can be varied through the rotating mirror (RM), such that for each input pattern a different tilt is applied to the mirror, resulting in a focal spot at different spatial positions at the CCD. Thus, the grating in PRC2 maps the input training set to distinct spatial spots at the output CCD.
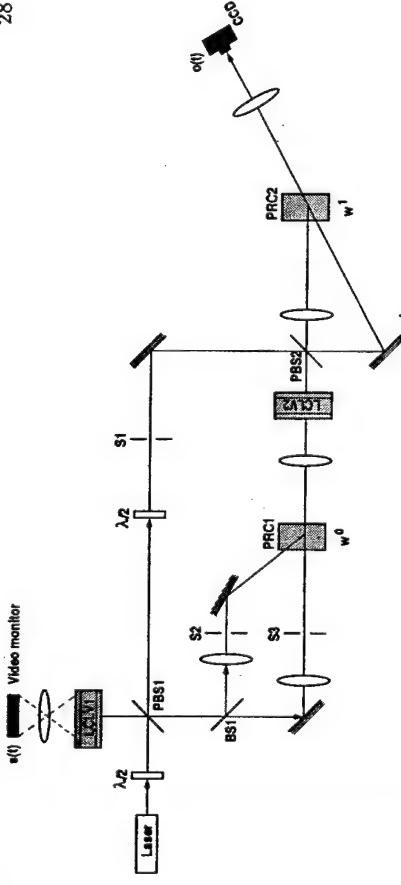
Figure 1.13: Psaltis and Qiao's [11] implementation of a 2-layer neural network. The first layer implements a dimensionality expanding random mapping, while the second layer maps the input patterns to a distinct spatial position in the output CCD. LCLV - liquid crystal light vale. PRC - photorefractive crystal. CCD - charge coupled device. BS - beam-splitter. PBS - polarizing BS. RM - rotating mirror.

Figure 1.14: Gan and Anderson's [12] implementation of a single-layer optical FIRNN with an associative learning algorithm. PCM - phase conjugate mirror. CCD - charge coupled device. SLM - spatial light modulator. M - mirror. L - lens. BS - beam splitter.

1.3.3 An optoelectronic implementation of an FIRNN with associative learning

Gan and Anderson [12] implemented a single-layer time-delay optical neural network making use of a rotating photorefractive crystal delay-line, as depicted in figure 1.14. This architecture makes use of a simple associative learning algorithm and it works as follows: the SLM modulates the light beam incident on the rotating PR crystal with the input information in one dimension. This causes a grating slice to be written in the volume hologram. Since the crystal c-axis is collinear with the rotation axis, previous versions of the hologram are simultaneously displayed in a conical pattern. A phase conjugate mirror (PCM) is used to read out the spatio-temporal plane, which writes a grating in the LiNbO$_3$ PR crystal in conjunction with the beam modulated by the second part of the SLM. This beam consists of a single SLM row, and each row is imaged onto a different spatial position at the output CCD. This way the PR grating performs a spatial mapping between time-delayed input patterns and spatial positions

at the output CCD. Notice that there is no feedback from the output to the input, making this architecture incapable of gradient descent learning. However, it works well for nearly orthogonal input patterns.

1.3.4 Optoelectronic implementations of adaptive antenna arrays

Several phased array optical architectures have been proposed [65, 66, 67, 68, 69, 70]. Though they are all capable of performing beam-forming, they are either not capable of processing temporally broadband signals, or are only capable of processing signals which are present in the far-field of a flat antenna array. This section discusses each one of these issues, highlighting the advantages of using an architecture based on a time-delay-and-sum single layer FIRNN with NT adaptive weights.

Transmit versus receive

Note that so far we have concentrated on adaptive antenna arrays used as re-

ceivers, where an adaptive algorithm steers the antenna gain pattern towards sources that are coherent with the desired signal. The same antenna arrays that are used for reception can also be used for transmission. If the weights are kept constant, the directivity pattern of the array used as a transmitter will be the same as the receptivity pattern of the same array used as a receiver.

Complex weights versus true-time-delay

Figure 1.15 shows an array processor that uses complex weights instead of TTD processing. Let a be the distance between antenna elements of a simple linear array with equidistant antenna elements. For simplicity of analysis, let us assume that the weights have already being set in order to receive a signal from the far-field, at an angle θ. This means that the weights must have a linear phase factor, i.e., $w_i = w_0 \exp(2\pi j\phi_i)$, where $\phi_i = \frac{\sin\theta ia}{v} f_0$, where v is the wave velocity, and f_0 is the frequency of the signal to be transmitted. Each antenna element will receive a signal $s_i(t) = e^{-i2\pi j\phi_i} S(t)$, where $S(t) = A\exp(j2\pi f_0 t)$ is a narrowband signal being received. Substituting, we have

$$s_i(t) = A w_0 \exp\left[2\pi j f_0 \left(\frac{ia\sin\theta}{v} + t\right)\right], \qquad (1.38)$$

so the phase shift acts as a time delay between antenna elements of $\Delta t = \frac{\sin\theta a}{v}$, a constant. However, now assume that a new signal $S'(t) = A\exp(j2\pi f_0' t)$ at a different frequency f_0' is received through the same set of weights. The signal at each antenna element now becomes

$$s_i'(t) = A w_0 \exp\left[2\pi j f_0' \left(\frac{f_0}{f_0'}\frac{ia\sin\theta}{v} + t\right)\right], \qquad (1.39)$$

and the new time delay between antenna elements is given by $\Delta t' = \frac{f_0}{f_0'}\frac{a\sin\theta}{v}$, which is also a constant, but this time it is a function of frequency. In effect, the new signal $S'(t)$ will be received from an angle given by $\theta' = \arcsin\left(\frac{f_0}{f_0'}\sin\theta\right)$, resulting in a shift of the peak of the antenna pattern with signal frequency. This effect is an undesired
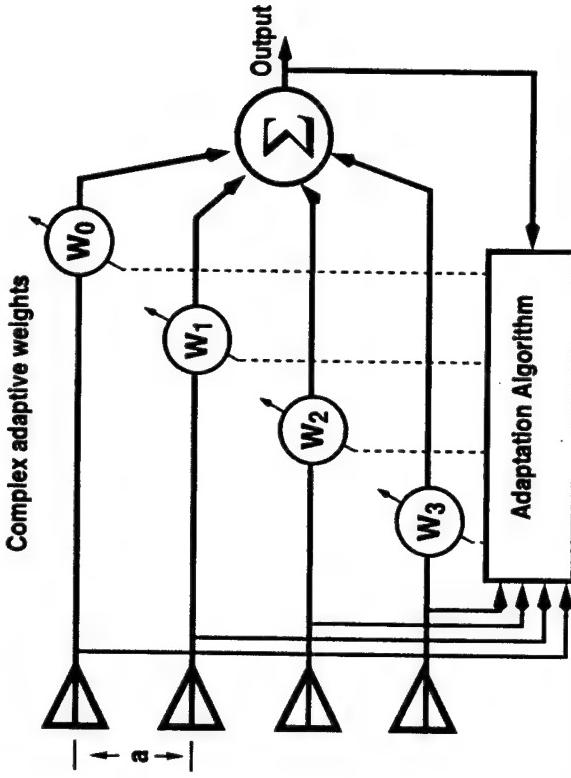
Figure 1.15: Adaptive array with complex weights instead of tapped-delay lines. Beam squint takes place in the presence of broadband signals .

limitation of antenna arrays which use complex weights for adaptation instead of TTD, and it is known as beam squint.

True-time delay ramps.

Figure 1.16 shows a TTD array processor which changes the array pattern by linearly incrementing the time-delay instead of the phase between antenna elements. It can be efficiently implemented by tuning the center wavelength of a laser diode which emits light through high-dispersion fiber in a binary-tree configuration [70]. In this case, even if $S(t)$ is broadband, the antenna element i will receive and delay the signal $s_i(t) = w_0 S(t - \tau_i)$, where $\tau_i = \Delta ti = \frac{ai\sin\theta}{v}$. Such an array processor can receive a signal from any one of T different directions without any beam squint. However, it is still incapable of dealing with problems which require more complex time-delay frunctions between antenna elements, such as in conformal arrays or near-field signal

Variable length delay lines
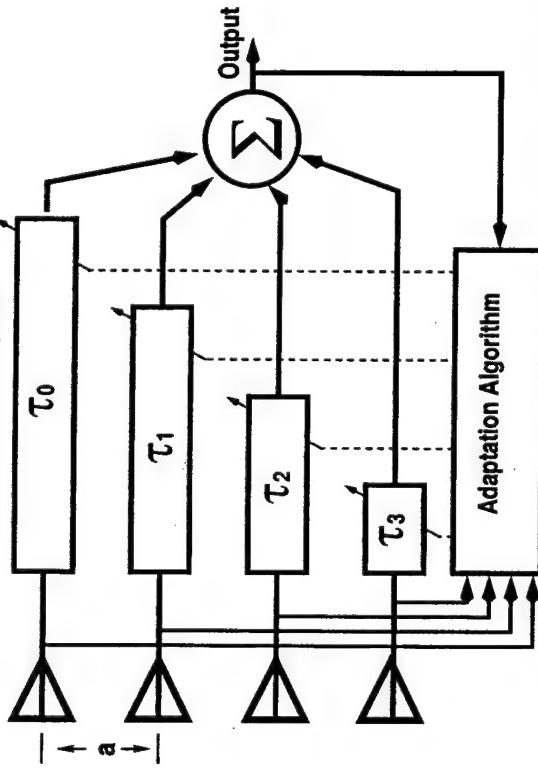
Figure 1.16: Adaptive array true time-delay processor with linearly increasing time-delays. It is capable of transmitting and receiving temporally broadband signals without beam-squint. However, it is only capable of processing signals that are spatially narrow-band, *i.e.*, present in the far-field of a flat antenna array.

detection/transmission. For example, in order to transmit a signal to the near-field, at a distance r from the antenna array, the time delay from the topmost antenna element to antenna element i has to be given by $t_i = \frac{\sqrt{r^2 - 2iar\cos\theta - i^2a^2}}{v}$, which is unachievable with a linearly increasing time-delay ramp.

Single-beam versus multiple beams

An N element adaptive antenna array with N complex weights, such as shown in figure 1.15, can steer N narrowband beams or nulls. However, the problems deriving from beam squint only allows one broadband beam or null to be formed. The broadband TTD beam steering system can only form one broadband beam and has no capability for nulling. To form multiple beams or nulls of broadband signals requires a TDL with

T taps for every antenna element. Such a TDL array can form N broadband beams or nulls simultaneously without beam squint.

Beam-forming versus adaptive processing and nulling

Finally, note that the linearly increasing time-delay architecture is capable of efficiently performing beam-forming in a given direction (provided that an efficient method exists for determining the desired direction). However, there are situations in which it is desirable to modulate the temporal spectrum of the processor. Such a situation exists, for example, in the presence of an in-band jammer, or when non-linearities are present (see chapter 4 for more details on non-linearities in the input signals of adaptive arrays). An adaptive processor requires added temporal degrees of freedom in order to be able to modulate its temporal response, such as provided by FIRNNs.

The single-layer FIRNN

All the previous limitations of the linearly increasing time-delay architecture can be overcome if a processor with N spatial and T temporal degrees of freedom is used, such as the ones based on the linear, single-layer FIRNN architecture (Fig. 1.6). Some optoelectronic implementations of this architecture make use of a photorefractive volume hologram for weight representation and a multi-channel acousto-optic Bragg cell [71, 72] for producing time-delayed versions of the input signal, providing the processor with full adaptability and all degrees of freedom necessary for broadband beam-forming and jammer nulling of signals, even in the near-field. However, they suffer from the limited number of channels that can be integrated in a single photoelastic crystal, which in practice is limited to a number between 32 and 64 [73].

Efficient FIRNN architectures

Two techniques for overcoming this limitation will be presented in this thesis. The

first consists of using the BEAMTAP method [6], which requires a single tapped-delay-line at the output instead of N TDLs in the input, as explained in chapter 2. The second method consists of using a 2-D SLM in a scrolling mode, performing a spatial mapping of the time-delayed versions of the input signals. Such a technique allows the processing of signals from 256 or more antenna elements, with 128 or more time-delays, for a total of 32768 degrees of freedom. This technique is explained in detail in chapter 3, and its experimental demonstration applied to a fully adaptive sonar processor is presented in chapter 5.

1.4 Summary

In conclusion, FIRNNs are found to posses definite advantages over traditional neural networks, especially in temporal prediction problems, control systems and adaptive array signal processing. Their real-time implementation is challenging, due to the high computational complexity of the operations involved; yet promising, since they comply with the basic requirements for successful optoelectronic implementations: the high complexity parallel linear operations can be easily separated from the lower order non-linear operations.

FIRNNs and its back-propagation algorithm are explained in more detail in chapter 2. Optical architectures for its implementation are discussed in chapter 4. Chapter 4 presents computer simulations exploring the application of single-layer FIRNNs as the learning algorithm for adaptive antenna arrays, including simulations of some of the experimental issues faced in the following chapter. Chapter 5 presents the analysis and experimental results of an implementation of a single-layer FIRNN designed for the adaptive broadband beam-forming and simultaneous multiple jammer nulling of sonar signals. Chapter 6 concludes this thesis, proposing future paths for extending the research work here presented.

Chapter 2

Finite Impulse Response Neural Networks

2.1 Introduction

This chapter describes in detail some of the algorithms used in the implementation of FIR neural networks. It starts by revisiting the FIR adaptive filter, the basic building block for FIRNN interconnects, and proceeds by describing the output signal generation in forward propagation. Wan's temporal back-propagation algorithm [2] is described next, followed by an alternative gradient descent learning algorithm, denoted the delayed-feedback back-propagation algorithm.

2.1.1 The FIR adaptive filter revisited

An FIR filter is a system whose output response $h(t)$ to an impulse input $\delta(t)$ is limited in time, that is, whose output is of the convolutional form $\sigma(\kappa) = \sum_\tau w_\tau \, s(\kappa-\tau)$, where $\sigma(\kappa)$ represents the filter output at time-step κ, $s(\kappa-\tau)$ represents the filter input τ steps before the current time-step κ, and the vector $\overline{w}$ represents the filter parameters. An adaptive FIR is a filter in which its parameters $\overline{w}$ may be changed in order to perform a specific task. Usually these parameters (the filter weights) are progressively adapted, following the prescription of a given learning algorithm, such as the LMS algorithm presented in chapter 1.

Figure 2.1 shows two diagrams illustrating two functionally equivalent representations of an FIR filter. The conventional one [Fig. 2.1(a)] consists of a tap-out delay-

$$\sigma(k)=\sum_{i=0}^{T} s(k-\tau)\, w_{\tau}(k)$$

(a)

$$\sigma(k)=\sum_{i=0}^{T} s(k-\tau)\, w_{\tau}(k-\tau)$$

(b)

Figure 2.1: Two equivalent representations of an FIR filter. (a) Traditional tap-out representation. (b) Tap-in, time-delay-and-sum representation.

line in which the delayed signal $s(\kappa - \tau)$ at tap τ is multiplied by the local weight w_τ. The result of each multiplication is summed, providing us with the final output $\sigma(\kappa) = \sum_{\tau=0}^{T} w_\tau(\kappa)\, s(\kappa - \tau)$. In the second representation [Fig. 2.1(b)] the input signal $s(\kappa)$ is simultaneously multiplied by all adaptive weights, and the partial products are then summed into the delay-line accumulator in a time-delay-and-sum operation, similar to that performed by a TDI detector, providing us with the final output $\sigma(\kappa) = \sum_{\tau=0}^{T} w_\tau(\kappa - \tau)\, s(\kappa - \tau)$. Notice that both representations are functionally identical, with the exception of the extra time-delay τ of the weights as they propagate through the TDL in the tap-in representation, which can be neglected when the weights are fixed or slowly varying. Also note that these representations suggest different physical implementations, as shall be seen in chapter 3.

2.1.2 Definition of the notation used in the mathematical description of FIRNNs.

In an FIRNN we have at each layer, for each time-delay of the TDL, one interconnecting weight between each input and output nodes, each one possibly varying at any given time step. As a consequence, the mathematical representation of FIRNNs require the use of tensors up to the fourth order and, for the sake of clarity, it is important to carefully define the notation used. The following conventions have been adopted in this thesis:

(1) **an upper bar over a vector represents a temporal dimension and a lower bar represents a spatial dimension**

The tensor $\underline{s}$, for example, represents a second rank tensor with a spatial and a temporal dimension

(2) **a quantity in parenthesis represents a time-step, indices represent specific vector elements**

This way, the tensor $\overline{w}_{ji}(\kappa)$ represents a temporal vector of FIR weights between nodes i and j at time-step κ. If a time-step is not indicated, the current time-step κ is assumed.

(3) **an upper index represents a specific network layer**

For example, vector $\underline{s}^l$ represents the input state vector at layer l, where $0 < l < L$, where L is the total number of layers in the network. Whenever necessary, the layer number may be enclosed in parenthesis in order to prevent confusion, e.g., $\underline{s}^{(1)}$ represents the input vector at layer 1. A new layer starts right after the output linear summation in forward propagation. This definition allows us to write N^l to denote the number of nodes in layer l.

(4) the first lower indices represent spatial elements while a lower index

after a comma represents a temporal element

Spatial and temporal indices are separated by a comma. For example, $w^l_{ji,\tau}(\kappa)$ represents the weight interconnecting nodes j and i at layer l after τ time delays in the TDL, at time-step κ.

Using these conventions, we define a time-series input signal at layer l and node i as $\overline{s}^l_i(\kappa) = [\, s^l_{i,\kappa} \;\; s^l_{i,\kappa-1} \;\; \cdots \;\; s^l_{i,\kappa-T^l} \,]^T$, where $\overline{s}^l_i(\kappa)$ represents the delayed states along the FIR interconnections departing from node i at layer l. T^l and N^l represent the total time delay and number of nodes at layer l. Analogously, the weights connecting layer i to layer j are defined as $\overline{w}^l_{ji} = [\, w^l_{ji,0} \;\; w^l_{ji,1} \;\; \cdots \;\; w^l_{ji,T^l} \,]^T$. Note that in forward propagation the states $\overline{s}^l_i(\kappa)$ are always a function of the current time-step κ, while the weights $\overline{w}^l_{ji}$ are not. The weights only become a function of the current time step when learning is also considered. Finally, for simplicity of notation, let us define $s^l_i(\kappa)$ (without the upper bar) as the element $s^l_{i,\kappa}$ of the vector $\overline{s}^l_i(\kappa)$.

2.2 The FIRNN in forward propagation

Figure 2.2 depicts a generic two-layer FIRNN ($L = 2$) in forward propagation.

Note that each FIR box represents an FIR filter as depicted in figure 2.1. In an FIRNN the input vector is always given by $\underline{s}^{(0)}(\kappa)$, and the final output vector is always given by $\underline{s}^{(L)}(\kappa) = \underline{o}(\kappa)$. The output from the FIR filter connecting node i to node j at layer l is given by σ^{l+1}_{ji}, and the sum of the outputs of all filters connected to node j is given by $\sigma^{l+1}_j = \sum_i \sigma^{l+1}_{ji}$. The non-linear discriminant function is denoted by $f(\cdot)$. We are now prepared to derive the mathematical representation of forward propagation in FIRNNs.

In forward propagation, a given input vector $\underline{s}^l$ is convolved with the adaptive weights present at each FIR interconnecting layer l to layer $l + 1$. The total signal
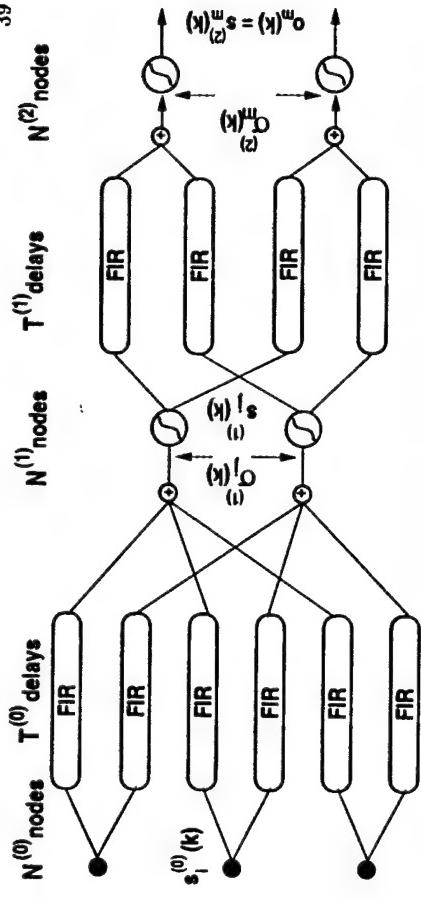
Figure 2.2: Schematic representation of a multi-layer finite impulse response neural network. FIR – finite impulse response adaptive filter interconnect. N^l – number of nodes in layer l. T^l – number of time delays in the TDL in layer l. $s^l_i(\kappa)$ – instantaneous input to node i, layer l. $\sigma^{l+1}_j(\kappa)$ – summation of FIR outputs to node j, layer $l + 1$.

arriving at node j (before the non-linear transformation) is given by the summation of the contribution from each individual FIR, i.e.,

$$\sigma^{l+1}_j(\kappa) = \sum_{i=1}^{N^l} \sum_{\tau=0}^{T^l} w^l_{ji,\tau}(\kappa)\, s^l_i(\kappa - \tau), \qquad (2.1)$$

After the non-linear operation $f(\cdot)$, the output of layer l is given by

$$s^{l+1}_j(\kappa) = f\left(\sum_{i=1}^{N^l} \sum_{\tau=0}^{T^l} w^l_{ji,\tau}(\kappa)\, s^l_i(\kappa - \tau) \right), \qquad (2.2)$$

which can be written in vectorial form as

$$\underline{s}^{l+1}(\kappa) = f\left(\underline{\underline{w}}^l : \overline{\underline{s}}^l(\kappa) \right), \qquad (2.3)$$

where the symbol : denotes a double dot product over the temporal dimension and the spatial dimension which $\underline{\underline{w}}$ and $\overline{\underline{s}}$ have in common (represented by the indices i and τ). Notice that equation 2.2 describes a spatio-temporal weighted sum followed by a non-linear function, where delayed versions of $\underline{s}^l(\kappa)$ are used, such that a windowed temporal convolution between $\underline{\underline{w}}$ and $\overline{\underline{s}}$ is performed in time for each of the N^{l+1} output

neurons. This operation consists of the point by point multiplication of the terms of two $N^l \times T$ matrices followed by the summation of all terms, performed N^{l+1} times, such that a total of $\sum_l N^{(l)} N^{(l+1)} T^{(l)}$ operations are performed at each time step.

2.3 On-line learning in FIRNNs

We are interested in FIRNN learning algorithms that make use of the gradient descent method in on-line learning, because of the greater flexibility they offer for potential real-time hardware implementations. An efficient algorithm which satisfies these properties is the temporal back-propagation learning algorithm [2], which will be derived next in the light of our mathematical notation. This derivation is followed by an interpretation of the behavior of the signals inside the delay-lines of an FIRNN during forward and temporal back-propagation. Such an interpretation is useful to the development of optoelectronic implementations of FIRNNs (Chapter 3), and to the understanding of alternative learning algorithms (Section 2.7).

2.3.1 Temporal back-propagation.

Note that in order to apply the gradient descent learning rule, it is necessary first to define a performance metric. Let us start by defining the instantaneous output error vector $\underline{e}(\kappa) = \underline{d}(\kappa) - \underline{s}^L(\kappa)$. The goal of learning is to make the network output be as close as possible to $d(\kappa)$, for all time. This can be accomplished by defining the following error metric:

$$C = \sum_{k=1}^{K} \| \underline{e}(\kappa) \|^2$$
$$= \sum_{\kappa=1}^{K} \sum_{j=1}^{N^L} (d_j(\kappa) - s_j^L(\kappa))^2, \quad (2.4)$$

where K is the number of data points in the training sequence $\underline{d}$, which could be infinite. According to the gradient descent learning rule, the weights are modified as

$$\overline{w}_{ji}^l(\kappa+1) = \overline{w}_{ji}^l(\kappa) - \eta \frac{\partial C}{\partial \overline{w}_{ji}^l}, \quad (2.5)$$

The first step to solve equation 2.5 is to expand the gradient on its right side using the chain rule for derivatives. Notice, however, that this equation defines the adaptation of a vector of weights, where each vector element represents a weight in a time-delay line. Since the cost function C involves a time summation, it is helpful to explicitly express the time-elements of equation 2.5 by using the less compact index notation, giving us

$$w_{ji,\tau}^l(\kappa+1) = w_{ji,\tau}^l(\kappa) - \eta \frac{\partial C}{\partial \sigma_j^{l+1}(\kappa)} \frac{\partial \sigma_j^{l+1}(\kappa)}{\partial w_{ji,\tau}^l}, \quad (2.6)$$

where σ_j^{l+1} is the input to the non-linear function of node j, going to layer $l+1$, defined in equation 2.1. The last derivative can be expanded, giving us

$$\frac{\partial \sigma_j^{l+1}(\kappa)}{\partial w_{ji,\tau}^l} = s_i^l(\kappa-\tau) \quad (2.7)$$

for any layer l. Therefore, our task now is reduced to the task of calculating the derivative $\frac{\partial C}{\partial \sigma_j^{l+1}(\kappa)}$. Following the conventional back-propagation algorithm [32], let us define

$$\delta_j^l(\kappa) = \frac{\partial C}{\partial \sigma_j^l(\kappa)}, \quad (2.8)$$

where $\delta_j^l(\kappa)$ represents the variation of the cost function due to a variation in the input to the non-linear function of node j, layer l, at time-step κ. This notation allow us to rewrite equation 2.5 as

$$\overline{w}_{ji}^l(\kappa+1) = \overline{w}_{ji}^l(\kappa) - \eta \delta_j^{l+1}(\kappa) \overline{s}_i^l(\kappa). \quad (2.9)$$

Now our task is to calculate δ_j^{l+1} for different network layers. Starting from the output layer and expanding equation 2.4, we have

$$\delta_j^L(\kappa) = \frac{\partial C}{\partial \sigma_j^L(\kappa)}$$

$$= \frac{\partial(e_j^2(1) + e_j^2(2) + \cdots + e_j^2(\kappa) + \cdots)}{\partial \sigma_j^L(\kappa)}$$

$$= \frac{\partial e_j^2(\kappa)}{\partial \sigma_j^L(\kappa)}. \tag{2.10}$$

Using equation 2.4, and noticing that $s_{j,k}^L = f(\sigma_j^L(\kappa))$, we have

$$\delta_j^L(\kappa) = \frac{\partial([d_j(\kappa) - f(\sigma_j^L(\kappa))]^2)}{\partial s_j^L(\kappa)} \quad \underbrace{}_{e_j(\kappa)}$$

$$= -2f'(\sigma_j^L(\kappa))[d_j(\kappa) - f(\sigma_j^L(\kappa))]$$

$$= -2f'(\sigma_j^L(\kappa))e_j(\kappa), \tag{2.11}$$

where $f'(\sigma_j^L)$ is the first derivative of $f(\sigma_j^L)$ with respect to σ_j^L. For a hidden-layer ($l < L$), we repeat this process using the chain rule [3], expanding the cost C over all time and over all N^{l+1} inputs $\sigma_m^{l+1}(\kappa)$ to the next layer

$$\delta_j^l(\kappa) = \frac{\partial C}{\partial \sigma_j^l(\kappa)}$$

$$= \sum_{m=1}^{N^{l+1}} \sum_{t=-\infty}^{+\infty} \frac{\partial C}{\partial \sigma_m^{l+1}(t)} \frac{\partial \sigma_m^{l+1}(t)}{\partial \sigma_j^l(\kappa)}. \tag{2.12}$$

Notice that $\partial C/\partial \sigma_m^{l+1}(t) = \delta_m^{l+1}(t)$. Substituting this expression in equation 2.12, expanding the second partial derivative using equation 2.1 and applying once again the chain rule, we get

$$\delta_j^l(\kappa) = \sum_{m=1}^{N^{l+1}} \sum_{t=-\infty}^{+\infty} \delta_m^{l+1}(t) \frac{\partial\left(\sum_{j=1}^{N^l}\sum_{\tau=0}^{T^l} w_{mj,\tau}^l s_{j,t-\tau}^l\right)}{\partial s_{j,k}^l} \frac{\partial s_{j,k}^l}{\partial \sigma_j^l(\kappa)}$$

$$= \sum_{m=1}^{N^{l+1}} \sum_{t-k=0}^{t-k=T^l} w_{mj,t-k}^l f'(\sigma_j^l(\kappa)) \delta_m^{l+1}(t). \tag{2.13}$$

The change in the time index in w takes place because the only non-zero terms of the first partial derivative are those terms where $t - \tau = k$, that is, where $\tau = t - k$. The time summation also has its indices changed. Instead of summing over all time, only the non-zero terms are considered. Rewriting equation 2.13, we have

$$\delta_j^l(\kappa) = f'(\sigma_j^l(\kappa)) \sum_{m=1}^{N^{l+1}} \sum_{t=k}^{k+T^l} w_{mj,t-k}^l \delta_m^{l+1}(t). \tag{2.14}$$

Changing the indices by defining $c = t - k$, we finally have

$$\delta_j^l(\kappa) = f'(\sigma_j^l(\kappa)) \sum_{m=1}^{N^{l+1}} \sum_{c=0}^{T^l} w_{mj,c}^l \delta_m^{l+1}(c+k). \tag{2.15}$$

Notice that equation 2.15 is non-causal, since it depends on future values of δ_j^l. The source of this non-causality is the definition of $\delta_j^l(\kappa) = \frac{\partial C}{\partial \sigma_j^l(\kappa)}$. δ_j^l measures the instantaneous effect of a specific node in the overall cost function. However, it takes a number of time-steps until that node propagates through the network and affects its output. Therefore, **the change in the total error due to a change in an internal state is a function of future values within the network.** Since the network interconnections are FIR filters, only a finite number of future time-steps must be considered. This can be achieved by re-indexing the previous equations, considering the time delay at each layer, giving us

$$w_{ji,\tau}^{L-1-n}(\kappa+1) = w_{ji,\tau}^{L-1-n}(\kappa) - \eta\delta_j^{L-n}(\kappa - \Upsilon^n) s_{i,k-\Upsilon^{n-\tau}}^{L-1-n}. \tag{2.16}$$

$$\delta_j^{L-n}(\kappa - \Upsilon^n) =$$

$$\begin{cases} -2e_j(\kappa) f'(\sigma_j^L(\kappa)) & n = 0 \\ f'(\sigma_j^{L-n}(\kappa - \Upsilon^n)) \sum_{m=1}^{N^{L-n+1}} \sum_{c=0}^{T^{L-n}} w_{mj,c}^{L-n} \delta_m^{L-n+1}(\kappa - \Upsilon^n + c) & 1 \le n \le L-1 \end{cases} \tag{2.17}$$

where Υ represents all the delays from the present layer to the network output and is defined by the following recursive expression

$$\Upsilon^n = \begin{cases} 0 & n = 0 \\ \Upsilon^{n-1} + T^{L-n} & 0 < n \le L. \end{cases} \tag{2.18}$$

Note that the last term of equation 2.17 is also a two-dimensional weighted sum, but this time it performs a temporal correlation between $\underline{w}$ and $\underline{\delta}$. Once again $\sum_l N^{(l)} N^{(l+1)} T^{(l)}$ operations per time step are required. These equations can also be more succinctly expressed in vectorial form, as follows:

$$\Delta\underline{w}^{L-1-n}(\kappa) = -\eta\,\underline{\delta}^{L-n}(\kappa - \Upsilon^n)\,\underline{s}^{L-1-n}(\kappa - \Upsilon^n), \tag{2.19}$$

where $\Delta \underline{\underline{w}}^{L-1-n}(\kappa) = \underline{\underline{w}}^{L-1-n}(\kappa+1) - \underline{\underline{w}}^{L-1-n}(\kappa)$, and

$$\underline{\delta}^{L-n}(\kappa - \Upsilon^n) = \begin{cases} -2f'(\underline{\sigma}^L(\kappa))\underline{e}(\kappa) & n = 0 \\ f'(\underline{\sigma}^{L-n}(\kappa - \Upsilon^n))\left[\underline{\underline{w}}^{L-n} : \underline{\overline{\delta}}^{L-n+1}(\kappa - \Upsilon^n)\right] & 1 \le n \le L-1 \end{cases}$$ (2.20)

That is, weight update can be described as an outer-product between the instantaneous delta terms and the time-delayed input. Back-propagation, on the other hand, can be described by the inner product of the derivative of the non-linear partial sums with the double dot product between the weights and the delayed delta terms, where the matrix $\underline{\overline{\delta}}^{L-n+1}(\kappa)$ is defined as having its elements with the most time delay listed first, that is,

$$\underline{\overline{\delta}}^{L-n+1}(\kappa - \Upsilon^n) = \begin{bmatrix} \underline{\delta}^{L-n+1}(\kappa - \Upsilon^{n+1}) \\ \underline{\delta}^{L-n+1}(\kappa - \Upsilon^{n+1} + 1) \\ \vdots \\ \underline{\delta}^{L-n+1}(\kappa - \Upsilon^n - 1) \\ \underline{\delta}^{L-n+1}(\kappa - \Upsilon^n) \end{bmatrix} .$$ (2.21)

2.4 Interpretation of the FIRNN back-propagation algorithm.

While equations 2.16 and 2.17 completely define the back-propagation algorithm used in FIRNNs, their interpretation is not trivial. Figure 2.3 proves to be helpful to this interpretation.

Figure 2.3 depicts the signal flow in the last layer of an FIRNN during forward and backward propagation. In forward propagation [part (a)] the input signals from the the previous layers are multiplied by the weights, summed and delayed, evaluating equation 2.2. Simultaneously, a derivative of the non-linear function with respect to the partial sums is generated. The output error is calculated at the output of the network. This term is multiplied by the learning rate and the non-linear derivative, producing the delta terms used to update the weights in the output layer. Part (b)
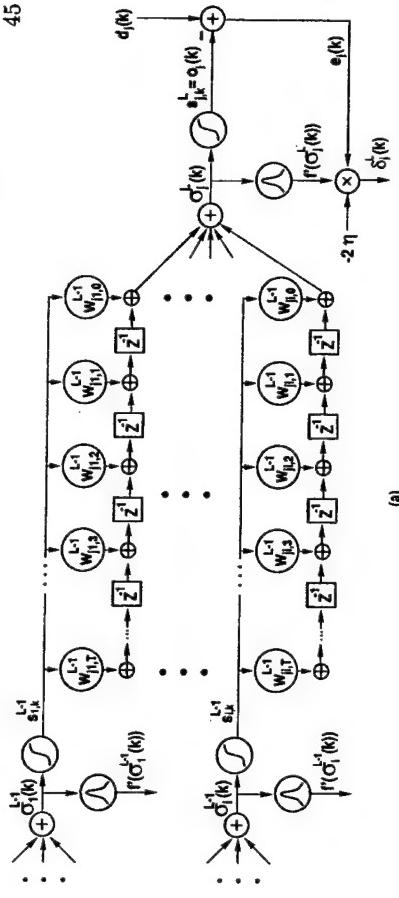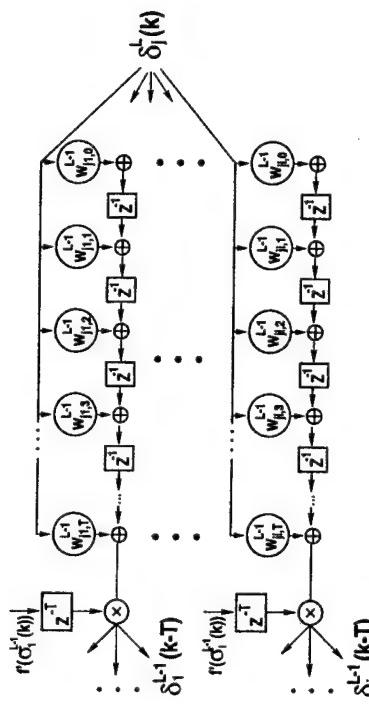
Figure 2.3: Signal flow in the last layer of an FIRNN during (a) forward propagation and (b) backward propagation. The weights shown in (a) and (b) are exactly the same, but signals flow in opposite direction in each case. Z^{-1} represents a discrete time delay.

shows backward propagation, where the delta terms are multiplied by exactly the same weights used in forward propagation, but delayed in reverse order to implement the temporal correlation. The output of each FIR is then multiplied by a delayed version of the non-linear derivative, producing the delta terms used to update the previous layer. The weights are then updated by an accumulation of the outer product between the back-propagated δ terms and a delayed version of the forward propagating input signal.

There are three important points to be observed:

(1) The order of the weights in back-propagation: the most recent δ terms (with the highest time-steps) are multiplied by the weights with the highest time indices. This is opposed to what happens in the forward propagation, where the state signals s with the lowest time indices are multiplied by the weights with the highest time indices.

(2) The back-propagated δ terms must then be multiplied by the time-delayed derivative of the FIR output sums $[f'(\underline{a}(\kappa - T))]$. That is because the instantaneous error at the current time-step is dependent on delayed versions of signal s. Note that this is a constant time-delay and, therefore, no extra TDL is necessary.

(3) The back-propagated δ terms are already in the form required to be used by equation 2.16: that is, back-propagation through n layers produces the term $\underline{\delta}^{L-n}(\kappa - \Upsilon^n)$, already delayed by Υ^n. No extra delaying of δ is necessary. However, $\underline{\delta}^{L-n-1}$ needs to be stored and delayed in a TDL. That is because equation 2.16 requires an outer-product between the vector $\underline{\delta}^{L-n}(\kappa - \Upsilon^n)$ and spatial-temporal matrix $\underline{s}^{L-n-1}(\kappa - \Upsilon^n)$.

2.5   Matrix representation

The implementation of an FIRNN and its back-propagation learning algorithm may become clearer using a matrix representation of its equations. Moreover, I found the following matrix equations a useful inspiration during the design of the optoelectronic architectures presented in chapter 3. Let us start with equation 2.3 describing the forward propagation operation. This equation requires the calculation of a point non-linear function of the summations of products between the weights w and the inputs s.

Expanding it, we have:

$$\begin{bmatrix} s_{1,k}^{l+1} \\ s_{2,k}^{l+1} \\ \vdots \\ s_{N^l+1,k}^{l+1} \end{bmatrix} = f \left(\sum_{\tau=0}^{T} \begin{bmatrix} \underline{w}_{1,\tau}^{l} \cdot \underline{s}_{k-\tau}^{l} \\ \underline{w}_{2,\tau}^{l} \cdot \underline{s}_{k-\tau}^{l} \\ \vdots \\ \underline{w}_{N^l,\tau}^{l} \cdot \underline{s}_{k-\tau}^{l} \end{bmatrix} \right), \quad (2.22)$$

Each element of equation 2.3 can be implemented as the non-linear function of a time-summation of dot products in order to implement the required convolutions. Notice that a dot product can also be implemented as the summation of all the elements of an inner-product, or as the summation of the diagonal elements of an outer-product. In optics, it could be implemented by the time integration of the sum of the spatial inner-products between $\underline{w}_\tau^l$ and $\underline{s}_{k-\tau}^l$. If we allow the temporal index to be represented in space, a space-integrating architecture could be used instead.

Now let us consider equation 2.17 when $n = 0$, that is, $\delta_j^{L}(\kappa) = -2e_j(\kappa) f'(\sigma_j^L(\kappa))$. This simple equation is expressed in matricial form as

$$\begin{bmatrix} \delta_1^L(\kappa) \\ \delta_2^L(\kappa) \\ \vdots \\ \delta_{N^L}^L(\kappa) \end{bmatrix} = -2 \begin{bmatrix} e_1(\kappa) f'(\sigma_1^L(\kappa)) \\ e_2(\kappa) f'(\sigma_2^L(\kappa)) \\ \vdots \\ e_{N^L}(\kappa) f'(\sigma_{N^L}^L(\kappa)) \end{bmatrix}, \quad (2.23)$$

which can be implemented as the point by point product of vectors $\underline{e}(\kappa)$ and $f'(\underline{\sigma}^L(\kappa))$.

Next, let's consider equation 2.17 for $1 \le n \le L - 1$.

$$\begin{bmatrix} \delta_1^{L-n}(\kappa - \Upsilon^n) \\ \delta_2^{L-n}(\kappa - \Upsilon^n) \\ \vdots \\ \delta_{N^{L-n}}^{L-n}(\kappa - \Upsilon^n) \end{bmatrix} = \begin{bmatrix} f'(\sigma_1^{L-n}(\kappa - \Upsilon^n)) \\ f'(\sigma_2^{L-n}(\kappa - \Upsilon^n)) \\ \vdots \\ f'(\sigma_{N^{L-n}}^{L-n}(\kappa - \Upsilon^n)) \end{bmatrix} \sum_{\tau=0}^{T^{L-n}} \begin{bmatrix} \underline{w}_{1,\tau}^{L-n} \cdot \underline{\delta}^{L+1-n}(\kappa - \Upsilon^n + \tau) \\ \underline{w}_{2,\tau}^{L-n} \cdot \underline{\delta}^{L+1-n}(\kappa - \Upsilon^n + \tau) \\ \vdots \\ \underline{w}_{N^{L-n},\tau}^{L-n} \cdot \underline{\delta}^{L+1-n}(\kappa - \Upsilon^n + \tau) \end{bmatrix}, \quad (2.24)$$

where $\underline{\delta}^l(\kappa)$ is defined as the following vector: $\underline{\delta}^l(\kappa) = [\ \delta_1^l(\kappa) \quad \delta_2^l(\kappa) \quad \cdots \quad \delta_{N^l}^l(\kappa)\ ]^T$. Notice that the right side of equation 2.24 is functionally equivalent to the spatio-

temporal dot-product found in equation 2.22 and, therefore, the same time and space-integrating methods can be used, followed by an inner-product with vector $f'(\underline{\alpha}^{L-n}(\kappa - \Upsilon^n))$.

Finally, equation 2.16 can be represented as

$$\begin{bmatrix} \Delta w_{11,t}^{L-n-1} & \Delta w_{21,t}^{L-n-1} & \cdots & \Delta w_{N^{L-n},1,t}^{L-n-1} \\ \Delta w_{21,t}^{L-n-1} & \Delta w_{22,t}^{L-n-1} & \cdots & \Delta w_{N^{L-n},2,t}^{L-n-1} \\ \vdots & \vdots & \ddots & \\ \Delta w_{1N^{L-n-1},t}^{L-n-1} & \Delta w_{2N^{L-n-1},t}^{L-n-1} & \cdots & \Delta w_{N^{L-n}N^{L-n-1},t}^{L-n-1} \end{bmatrix} =$$

$$= -\eta \begin{bmatrix} s_{1,k-\Upsilon^{n}-t}^{L-n-1} \\ s_{2,k-\Upsilon^{n}-t}^{L-n-1} \\ \vdots \\ s_{N^{L-n-1},k-\Upsilon^{n}-t}^{L-n-1} \end{bmatrix} \begin{bmatrix} \delta_1^{L-n}(\kappa-\Upsilon^n) & \delta_2^{L-n}(\kappa-\Upsilon^n) & \cdots & \delta_{N^{L-n}}^{L-n}(\kappa-\Upsilon^n) \end{bmatrix},$$

$$(2.25)$$

which is an outer-product between vectors $\underline{s}_{i,k-\Upsilon^{n}-t}^{L-n-1}$ and $\underline{\delta}^{L-n}(\kappa-\Upsilon^n)$, such as would be obtained from the temporal accumulation of the complex interference pattern between two spatially modulated beams when recording a dynamic volume hologram.

2.6 Linear FIRNNs and the LMS algorithm

The linear FIRNN is of special interest due to its ability to perform spatio-temporal adaptive linear filtering, while maintaining a relative simplicity of implementation. In such a case, any linear multi-layer network can be represented by a single-layer network, as long as the equivalent single-layer network has at least the same number of total time delays and sufficient adaptive weights. Under these circumstances, the FIRNN equations reduce to what is commonly known as LMS learning [5] (Fig. 2.4).

In a linear network, we note that $f(s(\kappa)) = s(\kappa)$ and $f'(s(\kappa)) = 1$. Furthermore, in a single-layer network, let $N^{(0)} = N$, $T^{(0)} = T$ and $\underline{g}^{(1)}(\kappa) = \underline{o}(\kappa)$. Thus, the output
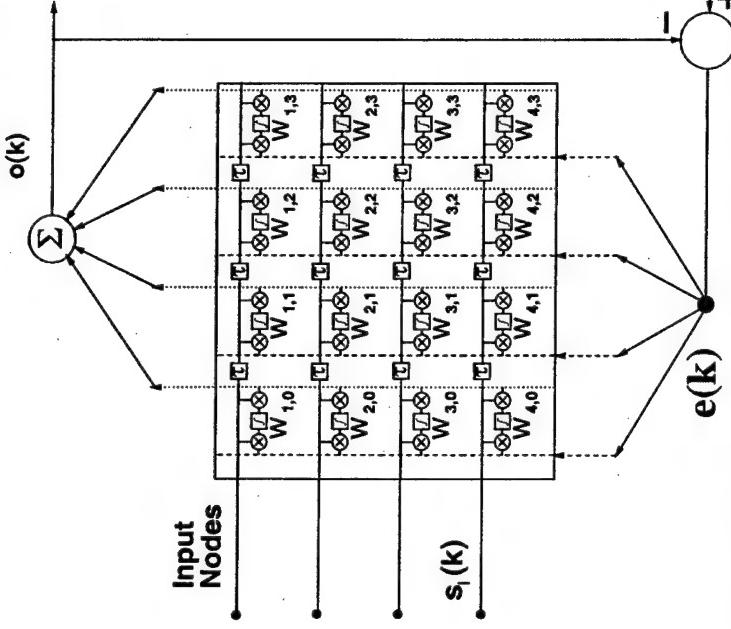
Figure 2.4: Conventional LMS single-layer FIRNN: each input node undergoes a delay-line. Each delayed input is multiplied by its respective weight and the partial products are summed at the output, which is subtracted from a desired signal producing an error signal. The error signal is locally multiplied by the delayed input and integrated at each weight.

is given by a single step of forward propagation

$$o_j(\kappa) = \sum_{i=1}^{N}\sum_{\tau=0}^{T} w_{ji,\tau}(\kappa)\, s_i(\kappa-\tau). \quad (2.26)$$

Similarly, using $L=1$, $n=0$ and $\Upsilon^n=0$, weight adaptation is given by

$$w_{ji,\tau}(\kappa+1) = w_{ji,\tau}(\kappa) - \eta\delta_j(\kappa)\, s_i(\kappa-\tau), \quad (2.27)$$

where $\delta_j(\kappa) = -2e_j(\kappa)$. Equation 2.27 requires the use of delayed versions of $s(\kappa)$ in order to adapt the weights. For a system with N inputs (from $s_1(\kappa)$ to $s_N(\kappa)$), this

would require the use of N tapped delay-lines, with $T - 1$ taps on each one of them, even for a single output node. This requirement suggests the need for an alternative method, capable of implementing FIRNNs with less delay-lines when a reduced number of output nodes is required.

2.6.1 BEAMTAP: a modification to the LMS algorithm

The Broadband and Efficient Adaptive Method for True-time-delay Antenna Processing (BEAMTAP) [6] algorithm is an alternative method for processing broadband signals in adaptive antenna arrays. The adaptive network architecture used by BEAMTAP consists of a single-layer linear FIRNN, but in which the input tapped delay-lines are replaced by delay-lines in the feedback path (Fig. 2.5). This makes it easier to implement the adaptive network with a space and time integrating optical architecture [74] that requires a smaller number of delay-lines whenever the number of output nodes is less than half the number of input nodes.

The BEAMTAP method consists of replacing the N tapped delay-lines in the input with delay-lines in the output and error signals. That is, we store delayed versions of the signal error $e_j(\kappa)$ and the partial sums $\sum_{i=1}^{q} w_{ji,\tau} s_i(\kappa - q)$ instead of the input $s_i(\kappa)$. To maintain causality, the delayed signal $s_i(\kappa - T)$ is used during weight update. This requires a simple constant delaying of the signal $s_i(\kappa)$. It is not necessary to access each time step and, therefore, no extra TDL is necessary at the input. Instead, 2 tapped delay-lines per output node are required, one operating as a conventional tap-out TDL, and the other operating as a time-delay-and-sum tap-in delay-line. Since a single-layer network usually has far less outputs than inputs, this method becomes interesting to implementations which require a small number of TDLs.

Applying the BEAMTAP method to a single-layer linear network, the weight update equation 2.27 becomes

$$w_{ji,\tau}(\kappa + 1) = w_{ji,\tau}(\kappa) + 2\eta e_j(\kappa - T + \tau) s_i(\kappa - T). \qquad (2.28)$$
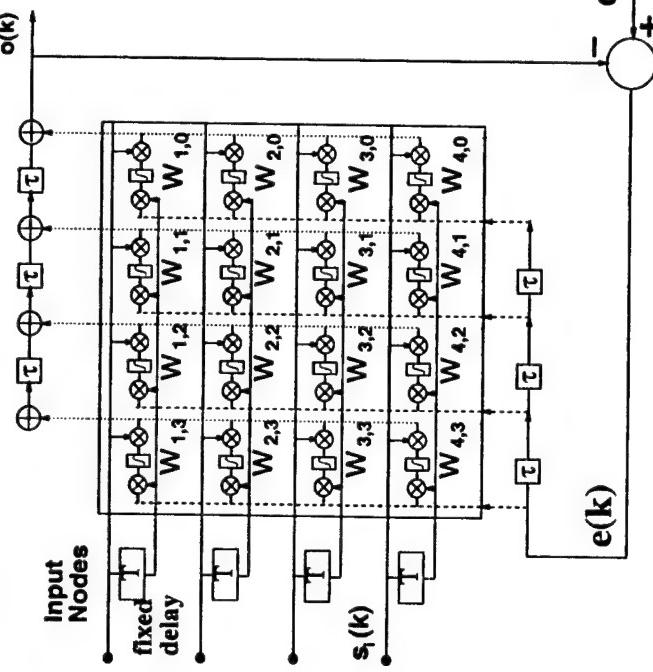
Figure 2.5: BEAMTAP version of a single-layer FIRNN: the inputs are simultaneously multiplied by all the weights at every time step. The resulting products are time delayed and summed at the output delay line, producing the final output, which is subtracted from a desired signal. The resulting error signal is delayed by another delay line and its delayed versions are multiplied by a delayed version of the input signal and locally integrated at each weight.

That is, the weights are adapted by an outer-product between delayed versions of the error and the input signal, with a fixed delay equals to the total amount of delay in the delay lines. In forward propagation, the final expression is still a spatial-temporal dot-product, such as depicted in equation 2.3. The delay line in the output results in the use of delayed versions of the weights, as described by the following equation:

$$o_j(\kappa) = \sum_{i=1}^{N}\sum_{\tau=0}^{T} w_{ji,\tau}(\kappa - \tau) s_i(\kappa - \tau). \qquad (2.29)$$

We are assuming that the weights are updated at every time step. In this case, the instantaneous output is observed to be a function of delayed versions of the weights.

This is a consequence of delaying the partial sums in order to generate the output and does not impose any problem if the input signals are wide-sense stationary (WSS) or if the weights are slowly varying $\underline{w}(\kappa - T) \cong \underline{w}(\kappa)$].

2.7 The delayed-feedback back-propagation algorithm

The BEAMTAP method described above can be extended to multi-layer, non-linear FIRNNs. The modified algorithm is called **delayed-feedback back-propagation** [75], in contrast with Wan's temporal back-propagation [2]. At each layer l the tapped delay-lines are moved from the input signals $s_i^l(\kappa)$ to the error signals $\overline{\delta}_Q^{l+1}(\kappa - \Upsilon^{L-l+1})$ and partial sums $\sum_i w_{ji,\tau}^l s_i^l(\kappa)$. In forward propagation the net effect is analogous to the case of using a time-delay-and-sum representation of an FIR filter [Fig. 2.1(b)]. As we have already seen, the FIR output is functionally equivalent to the conventional tap-out case. Consequently, the forward propagation equation is similar to that obtained for conventional FIRNNs, with the exception that the weights are also delayed as the partial products propagate through the TDL, i.e.,

$$s_j^{l+1}(\kappa) = f\left(\sum_{i=1}^{N}\sum_{\tau=0}^{T} w_{ji,\tau}^l(\kappa - \tau)\, s_i^l(\kappa - \tau)\right). \quad (2.30)$$

Using the same definition of a cost function in equation 2.4 and following the gradient descent rule, weight update can be expressed in terms of partial derivatives of delayed versions of the partial sums, as follows

$$w_{ji,\tau}^l(\kappa + 1) = w_{ji,\tau}^l(\kappa) + \Delta w_{ji,\tau}^l(\kappa), \quad (2.31)$$

with

$$\Delta w_{ji,\tau}^l(\kappa) = -\eta \frac{\partial C}{\partial w_{ji,\tau}^l(\kappa)}$$
$$= -\eta \frac{\partial C}{\partial \sigma_j^{l+1}(\kappa - \xi^l)}\,\frac{\partial \sigma_j^{l+1}(\kappa - \xi^l)}{\partial w_{ji,\tau}^l(\kappa)}, \quad (2.32)$$

where ξ^l represents an arbitrary integer time delay. To simplify our derivation, let us define

$$\delta_j^l(\kappa - \xi^l) = \frac{\partial C}{\partial \sigma_j^l(\kappa - \xi^l)}, \quad (2.33)$$

as the delayed back-propagation term. Expanding the second term of equation 2.32, we have

$$\frac{\partial \sigma_j^{l+1}(\kappa - \xi^l)}{\partial w_{ji,\tau}^l} = \frac{\partial\left(\sum_{j=1}^{N^l}\sum_{\tau=0}^{T^l} w_{ji,\tau}^l\, s_i^l(\kappa - \tau - \xi^l)\right)}{\partial w_{ji,\tau}^l} =$$
$$= s_i^l(\kappa - \tau - \xi^l). \quad (2.34)$$

Now let us expand the term $\delta_j^l(\kappa - \xi^l)$, starting with the last layer ($l = L$) and using $C = \sum_k \sum_j e_j^2(\kappa)$.

$$\delta_j^L(\kappa - \xi^l) = \frac{\partial C}{\partial \sigma_j^L(\kappa - \xi^l)}$$
$$= \frac{\partial e_j^2(\kappa - \xi^l)}{\partial \sigma_j^L(\kappa - \xi^l)},$$

where we assumed $e(\kappa)$ to be zero for $\kappa < 0$. Expanding the error term $(e_j(\kappa) = d_j(\kappa) - s_j^L(\kappa))$ and using $s_j^L(\kappa) = f(\sigma_j^L(\kappa))$, we have

$$\delta_j^L(\kappa - \xi^l) = \frac{\partial\left(\overbrace{[d_j(\kappa - \xi^l) - f(\sigma_j^L(\kappa - \xi^l))]}^{e_j(\kappa-\xi^l)}{}^2\right)}{\partial \sigma_j^L(\kappa - \xi^l)}$$
$$= -2f'(\sigma_j^L(\kappa - \xi^l))\,[d_j(\kappa - \xi^l) - f(\sigma_j^L(\kappa - \xi^l))]$$
$$= -2f'(\sigma_j^L(\kappa - \xi^l))\,e_j(\kappa - \xi^l). \quad (2.36)$$

For a hidden-layer ($l < L$), we repeat this process expanding the cost C over all time and over all N^{l+1} inputs $\sigma_m^{l+1}(\kappa - \xi^l)$ to the next layer

$$\delta_j^l(\kappa - \xi^l) = \sum_{m=1}^{N^{l+1}}\sum_{t=-\infty}^{+\infty} \frac{\partial C}{\partial \sigma_m^{l+1}(t - \xi^l)}\,\frac{\partial \sigma_m^{l+1}(t - \xi^{l+1})}{\partial \sigma_j^l(\kappa - \xi^{l+1})}$$
$$= \sum_{m=1}^{N^{l+1}}\sum_{t=-\infty}^{+\infty} \delta_m^{l+1}(t - \xi^{l+1})\,\frac{\partial\left(\sum_{j=1}^{N^l}\sum_{\tau=0}^{T^l} w_{mj,\tau}^l\, s_j^l(t - \tau - \xi^l)\right)}{\partial s_j^l(\kappa - \xi^l)}\,\frac{\partial s_j^l(\kappa - \xi^l)}{\partial \sigma_j^l(\kappa - \xi^l)}, \quad (2.37)$$

where we made the substitution $\partial C/\partial \sigma_m^{l+1}(t - \xi^l) = \delta_m^{l+1}(t - \xi^l)$. Evaluating the two partial derivatives and performing the time summation only over the non-zero terms, we have

$$\delta_j^l(\kappa - \xi^l) = \sum_{m=1}^{N^{l+1}} \sum_{t=\kappa}^{\kappa+T^l} w_{mj,t-\kappa}^l \delta_m^{l+1}(t - \xi^{l+1}) f'(s_j^l(\kappa - \xi^l))$$

$$= f'(\sigma_j^l(\kappa - \xi^l)) \sum_{m=1}^{N^{l+1}} \sum_{c=0}^{T^l} w_{mj,c}^l \delta_m^{l+1}(\kappa + c - \xi^{l+1}), \qquad (2.38)$$

where we defined $c = t - \kappa$.

Now we are able to define the time delay ξ^l. Equation 2.38 describes the back-propagation of the error $\overline{e}^{l+1}$ term from layer to layer. Such an operation can be implemented using a tapped delay-line with T^l delays in which the terms δ^{l+1} are fed one after another at each time step. Consequently, the latest term of the delay line will be delayed by T^l steps with respect to the first one and the back-propagated terms δ^l would be delayed by T^l with respect to the terms in the previous layer (δ^{l+1}). Thus, in order to maintain causality, it is necessary to delay the term $\delta_m^{l+1}(\kappa + c - \xi^{l+1})$ in equation 2.38 by at least Υ^{L-l+1} time steps (the total time delay from the input of layer l to the network final output, defined in equation 2.18). Finally, equation 2.34 can be further simplified by defining ξ^l as

$$\xi^l = \Upsilon^{L-l+1} - \tau, \qquad (2.39)$$

and we are able to use the minimum number of time delays while simultaneously satisfying the principle of causality and simplifying the term $s_i^l(\kappa - \tau - \xi^l)$ to $s_i^l(\kappa - \Upsilon^{L-l+1})$.

That is, a fixed amount of time delay is now required for each input, instead of a tapped delay-line. Defining $n = L - l$ (the layer number counting from the output), we arrive at the following learning algorithm for temporal back-propagation:

$$\Delta w_{ji,\tau}^{L-n-1}(\kappa) = -\eta \, \delta_j^{L-n}(\kappa - \Upsilon^{n+1} + \tau) \, s_i^{L-n-1}(\kappa - \Upsilon^{n+1}) \qquad (2.40)$$

where

$$\delta_j^{L-n}(\kappa - \Upsilon^{n+1} + \tau) =$$

$$\begin{cases} -2f'(\sigma_j^L(\kappa - \Upsilon^{L-1} + \tau)) \, e_j(\kappa - \Upsilon^{L-1} + \tau) & n = 0 \\ f'(\sigma_j^{L-n}(\kappa - \Upsilon^{n+1} + \tau)) \sum\limits_{m=1}^{N^{L-n+1}} \sum\limits_{c=0}^{\Upsilon^{L-n}} w_{mj,c}^{L-n} \delta_m^{L-n+1}(\kappa - \Upsilon^{n+1} + c + \tau) & 1 \le n \le L-1 \end{cases} \qquad (2.41)$$

Note that the temporal correlation between $w_{jm,c}^{L-n}$ and $\delta_m^{L-n+1}(\kappa - \Upsilon^{n+1} + \tau)$ is still present, but this time the delta term has another temporal index, τ. Again, this calculation requires $N^{(l)}N^{(l+1)}T^{(l)}$ operations, as does weight update. The back-propagation term $\delta_j^{L-n}(\kappa - \Upsilon^{n+1} + \tau)$ must be delayed in a TDL, so that it can be used in weight update and also be back-propagated. Finally, during weight update the input vector $\underline{s}^{L-n-1}$ must be delayed by a fixed amount of time steps (Υ^{n+1}), equal to the number of time delays from layer $L - n$ to the output, but it requires no tapped delay-line.

In vector notation, we have

$$\Delta \underline{\overline{w}}^{L-1-n}(\kappa) = -\eta \, \underline{\overline{\delta}}^{L-n}(\kappa - \Upsilon^n) \, \underline{s}^{L-1-n}(\kappa - \Upsilon^n), \qquad (2.42)$$

where $\underline{\overline{\delta}}^{L-n}(\kappa - \Upsilon^n)$ is defined as in equation 2.20 and $\underline{\overline{\delta}}^{L-n}(\kappa - \Upsilon^n)$ is defined as in equation 2.21. That is, the only difference is that we are using delayed versions of the delta error term instead of the input. The other changes in index are present in order to assure synchrony and causality.

2.8 Summary

In this chapter FIR adaptive filters and FIRNNs have been presented under different forms and some of their applications have been listed. A mathematical formalism based on multiple tensorial indices was created in order to facilitate the analyses of the forward and back-propagation equations that describe FIRNNs. Wan's efficient temporal back-propagation algorithm was presented and analyzed. Single-layer, linear FIRNN have been presented and their analogy with the LMS algorithm was shown. A modifi-

cation of the LMS algorithm was presented, known as the BEAMTAP method, which relies on a topological rearrangement of the TDLs in the network, using a single tap-in and a tap-out delay line instead of a TDL per input channel. This provided inspiration for the extension of the same principle to multi-layer, non-linear FIRNNs, creating an algorithm known as delayed-feedback back-propagation.

In the next chapter we will see that the conventional back-propagation and the delayed feedback back-propagation algorithms are functional duals of each other. As such, either one of them can be used in optical architectures making use of similar physical devices with equivalent parameters, the main difference being the physical location of the tap-in or tap-out delay-lines.

Chapter 3

Optoelectronic architectures for Finite Impulse Response Neural Networks

3.1 Introduction

In chapter 2, Finite Impulse Response Neural Networks have been presented and analyzed. FIRNNs, like most neural network architectures, can take advantage of the parallelism and capacity for arbitrary global interconnects that an optical implementation offers. However, they have the additional requirement of processing information in the time domain. As will be shown in the current chapter, this requirement can be dealt with by storing delayed versions of either the input or the output of the FIR neural network, performing what shall be called the **input delay plane** or the **output delay plane** architectures. Based on these architectures, single-layer and multi-layer FIRNN architectures are proposed and briefly analyzed making use of the temporal back-propagation and the delayed-feedback back-propagation algorithms.

Let us begin by searching for optical architectures capable of efficiently performing the spatial-temporal dot products, since these are the most computationally demanding operations required in the implementation of FIRNNs, and their efficient implementation is critical in forward and back-propagation, in the conventional and in the delayed-feedback algorithms.

Figure 3.1: Basic topology of the input delay plane architecture using a scrolling SLM and space integration in a row of output detectors. Δ_t denotes the time delay between time-steps.

input row, such that the input signal at channel i, delayed by τ time-steps, gets spatially mapped into the ith column, τth row of the SLM. A volume hologram performs the weighted interconnection from the input plane to the array of output detectors, spatially integrating the partial products $s_i(\kappa - \tau)\,w_{ji,\tau}$, producing the output signal $\underline{\sigma}(\kappa)$. The signal detected at the jth output detector is given by the summation of all the input signals interconnected from the input plane to this node, that is,

$$\sigma_j(\kappa) = \sum_i s_{i,\kappa} w_{ji,0} + \sum_i s_{i,\kappa-1} w_{ji,1} + \cdots + \sum_i s_{i,\kappa-T} w_{ji,T}$$
$$= \sum_{\tau=0}^{T} \sum_{i=1}^{N} s_{i,\kappa-\tau}\, w_{ji,\tau} \qquad (3.2)$$

which is in agreement with equation 2.3 for forward propagation under the temporal back-propagation algorithm.

A scrolling SLM can be implemented by several different technologies, each with

3.2 The spatial-temporal dot product

The spatial-temporal dot product is an operation of the form

$$A_j(\kappa) = \sum_\alpha \sum_\tau w_{ji,\tau} B_{\alpha,\tau}, \qquad (3.1)$$

where $A = a^{l+1}$, $\overline{B} = \overline{s}^l$ and $\alpha = i$ in forward propagation, and $A = \delta^l/f'(a^l)T^{(l)}$, $\overline{B} = \overline{a}^{l+1}$ and $\alpha = j$ in back-propagation. This operation requires $N^{(l)}N^{(l+1)}T^{(l)}$ steps and, therefore, has a cubic order of computational complexity. Thus, as the number of nodes and time-delays in the network increases, the amount of computation required grows very rapidly. It is possible, however, to implement this linear operation optically such that the time required to perform the calculation is fixed and the maximum number of nodes and time-delays is limited only by the resolution of the hardware employed in the implementation. As seen in chapter 2, the delays in an FIRNN can be in the form of a tap-out delay-line applied to each of the input channels (in conventional back-propagation), or in the form of a tap-in delay-line applied to each of the partial sum of products in the output and another tap-out delay applied to each of the feedback error signals (in delayed-feedback back-propagation). Each one of these algorithms suggests a different optoelectronic architecture. For simplicity of notation, let us assume for now that the spatial-temporal dot product is being applied to the case of forward propagation, since back-propagation is only interesting in the more complex case of multi-layer neural networks.

3.2.1 Input delay plane architecture

The input delay plane architecture makes use of tap-out delay-lines [see Fig. 2.1(a)]. N TDLs are used in parallel in order to implement the spatio-temporal dot product, as depicted in figure 3.1. This architecture is based on a scrolling spatial light modulator with T rows and N columns, and operates as follows: at each time-step, each row of the scrolling SLM shifts up one row and a new version of the input vector $\underline{s}(\kappa)$ modulates its

its own advantages. The best choice depends on the desired number of input channels, number of time-delays, speed, contrast ratio, and cost. The first example are multi-channel AO Bragg cells [76], which have a time-bandwidth product up to 1000 at each channel and operate at a center frequency of about 100 MHz, but they are limited to a number of input channels between 32 and 64 [73]. In addition, the cost of the RF electronic circuitry required to drive each channel increases proportionally to the number of channels.

A second option is to use an array of multiple quantum-well (MQW) modulators [77, 78]. These devices make use of the quantum-confined Stark effect [79, 80] to change the absorption of the modulator by moving the center wavelength of the exciton absorption peak with respect to an applied electric field. As a consequence, these devices are very fast, operating at data rates higher than 1 Gb/s. Their structure is based on GaAs quantum-wells, but they can be flip-chip bonded onto a complementary metal-oxide semiconductor (CMOS) substrate [78], greatly simplifying the design of the underlying driving and shifting electronics and reducing the cost for a large number of channels. However, these devices are limited by the small movement of the exciton peak that can be achieved by the maximum voltage swing in a semi-conductor, limiting the maximum contrast ratio to about 10 to 1.

A third option is to use a rotating dynamic volume hologram, like the one developed by Gan and Anderson [81]. Their device is based on a rotating PR crystal which is illuminated by a reference beam collinear with the rotation axis of the crystal (and the crystal axis), as shown in figure 3.2(a). A temporally modulated 1-D beam array containing the signal $s(\kappa)$ is set to interfere with the reference beam. The two beams continually record holographic gratings in the crystal as it rotates at an angular velocity $\dot{\theta}$. Since the reference beam is collinear with the axis of rotation, it is always Bragg matched with the previously recorded gratings. As a consequence, the delayed signal $s(t - t')$ is spatially mapped into a cone of simultaneously diffracted holograms, where
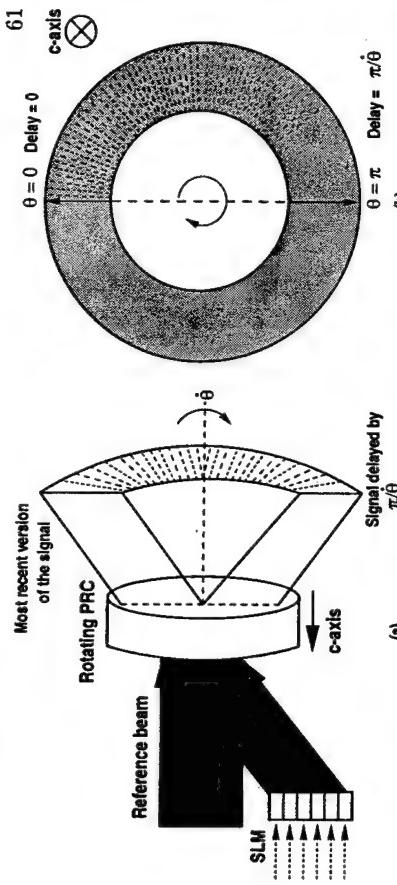
Figure 3.2: Operation of a rotating PRC as a scrolling SLM. (a) Reference and signal beams interfering in a rotating PRC and producing delayed versions of the signals in a circular segment of its output cone. (b) Output circle showing the progression of the delay with rotation angle.

the delay is proportional to the azimuth angle, i.e., $t' = \theta/\dot{\theta}$ [Fig. 3.2(b)]. Photorefractive decay causes the signal amplitude to decay as it goes around the cone, providing us with a signal proportional to $s(t - t')e^{-t'/\tau_{pr}}$. Once the PR crystal undergoes a complete revolution ($\theta = 2\pi$), a new grating is recorded over the previous ones, providing us with a continuously scrolling SLM. This elegant approach, however, is limited by the maximum speed of rotation of the PRC.

A more straightforward solution is to use electronically addressed liquid crystal (LC) SLMs [82]. A nematic LC is capable of grey scale modulation, but it is very slow (only up to 100 Hz switching frequency). A ferro-electric liquid crystal (FLC) SLM operates up to 5 kHz. However, it is only capable of binary data representation. This limitation can be compensated by using a high resolution device. Figure 3.3 shows the resulting image when a 256 × 256 FLC-SLM is fed with a binary chirp with frequency varying linearly from 93.8 Hz to 563 Hz at a 1.50 kHz sampling frequency. All input nodes receive the same signal, but with a delay of 0.18 ms between nodes, which causes a spatial shift of the resulting pattern. Figure 3.4 shows the Fourier plane of this
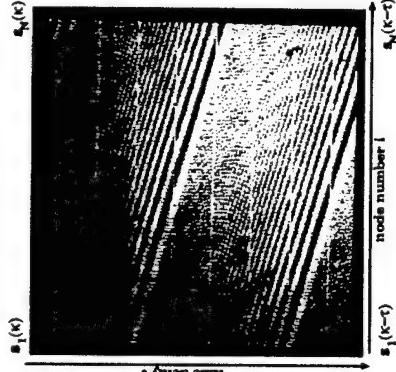
Figure 3.3: SLM frame captured in the image plane, displaying a continuous broadband binary chirp with linear delay between input channels.

Figure 3.4: Fourier plane of the SLM, displaying a long and thin line with constant tilt, showing that the signal is temporally broadband, repetitive, with a linear delay between SLM columns.

modulated image. It is a long and thin line, telling us that the signal is temporally broadband, but spatially narrowband. Chapter 5 reports in detail the use of such a device in an input delay plane architecture.

3.2.2 Output delay plane architecture

The output delay plane architecture makes use of N tap-in delay-lines in parallel, such as the one depicted in figure 3.5(a). It operates as follows: a one-dimensional SLM is used to modulate the input vector with signal $s(\kappa)$. At each time-step, the input vector is diffracted by an interconnecting volume hologram, and the resulting partial products $w_{ij,\tau} s_i(\kappa)$ are spatially mapped into pixel positions (j,τ) of the output plane [Fig. 3.5(b)]. A TDI photodetector is placed at this plane, oriented so that each one of the (j,τ) output pixels corresponds to a photon detection site. Each site spatially integrates all the partial products detected by it and, at each time-step, the TDI detector scrolls down the signals detected at each one of its T rows and detects another set of partial products in a time-delay-and-sum operation.
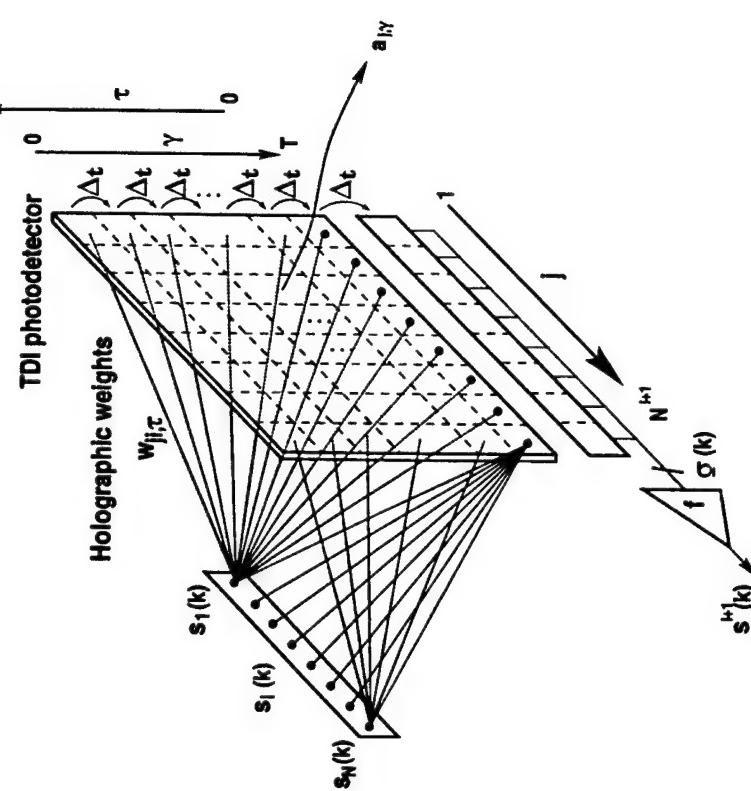
Figure 3.5: Basic topology of the output delay plane architecture using a holographic interconnection followed by space and time integration in a TDI photodetector. τ and γ are complementary temporal indices.

To illustrate this operation, let γ be a spatial index varying from zero to T in the direction of temporal shift of the TDI detector, such that $\gamma = T - \tau$, and let the signal accumulated at the photon site (j,γ) be $a_{j,\gamma}(\kappa) = a_{j,\gamma-1}(\kappa) + \sum_i w_{ij,\tau-\gamma}(\kappa)\, s_i(\kappa)$ (i.e., the accumulation of previously detected charge carriers shifted to this site, plus the spatial integration of the partial products being currently detected). Expanding the recurrent summation, we have

$$a_{j,\gamma}(\kappa) = \sum_{c=0}^{\gamma}\sum_{i=1}^{N} w_{ji,T-c}(\kappa-\gamma+c)\, s_i(\kappa-\gamma+c), \qquad (3.3)$$

where c is an auxiliary index, varying from 0 to γ. At the TDI detector output, $\gamma = T$,

and we have

$$a_{j,T}(\kappa) = \sum_{c=0}^{T}\sum_{i=1}^{N} w_{ji,T-c}(\kappa-T+c)\, s_i(\kappa-T+c)$$

$$= \sum_{\tau=0}^{T}\sum_{i=1}^{N} w_{ji,\tau}(\kappa-\tau)\, s_i(\kappa-\tau)$$

$$= \sigma_j(\kappa), \qquad (3.4)$$

where the substitution $\tau = T - c$ was made. Note that equation 3.4 is equivalent to the equation for forward propagation under the delayed-feedback algorithm (Eqn. 2.30).

3.3 Holographic interconnections and bipolar signal representation

Both the input delay plane and the output delay plane architectures presented in the previous section utilize holographic interconnections between a plane with TN pixels and a line with N pixels for representing the adaptive weights. This section explains in more detail how volume holograms can perform such a task and discusses other relevant issues, such as the effect of Bragg degeneracy in each architecture and different methods for bipolar signal representation.

3.3.1 Holographic interconnections for the input delay plane and the output delay plane architectures

In section 1.2.2 we have seen how a volume hologram can perform weighted interconnections between two plane waves. Let us now extend this principle to a generic optical system that illustrates the operation of holographic interconnections for both the input delay plane and the output delay plane architectures, shown in figure 3.6. A 2-D input modulator with NT pixels and a 1-D input modulator with N pixels modulate the input beams which interfere and record a volume grating in the central PR crystal placed in their Fourier planes. The system analysis using $\bar{k}$-space is simplified in this case, but the conclusions are valid even when the crystal is not at the Fourier plane but
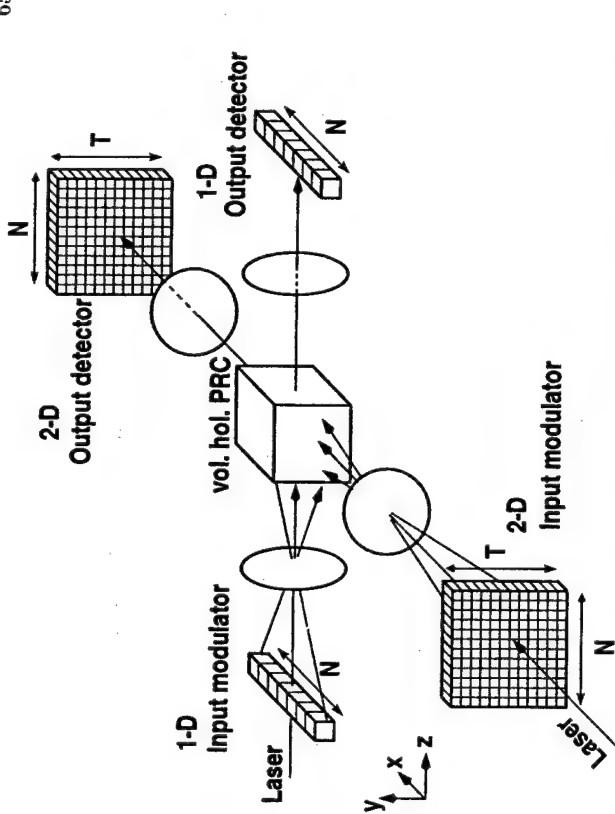
Figure 3.6: Generic architecture illustrating the holographic interconnections between a 2-D plane and a 1-D vector, present in the input delay plane and output delay plane architectures.

at an image plane or at an arbitrary Fresnel plane. An operator-notation analysis of a generic version of this system is presented in appendix A. A 4-f system exactly images the input 1-D and 2-D modulators pixel by pixel towards the output 1-D and 2-D detectors. Each input pixel can be modeled as a point source which, in the Fourier plane, produces a plane wave with a specific wave vector. Figure 3.7 shows the input waves and the resulting holographic gratings in $\bar{k}$-space. For clarity, only the central wave vectors (i.e., the wave vector corresponding to the central pixels) are represented, given by $\bar{k}_{1-D}$ and $\bar{k}_{2-D}$. The wave vectors produced by the other pixels are represented by small gray circles corresponding to the position of their tips on the wave vector surface. The total grating is given by the superposition of all the gratings recorded between each input pixel pair, for a total of N^2T gratings, using all of the three dimensions available in the
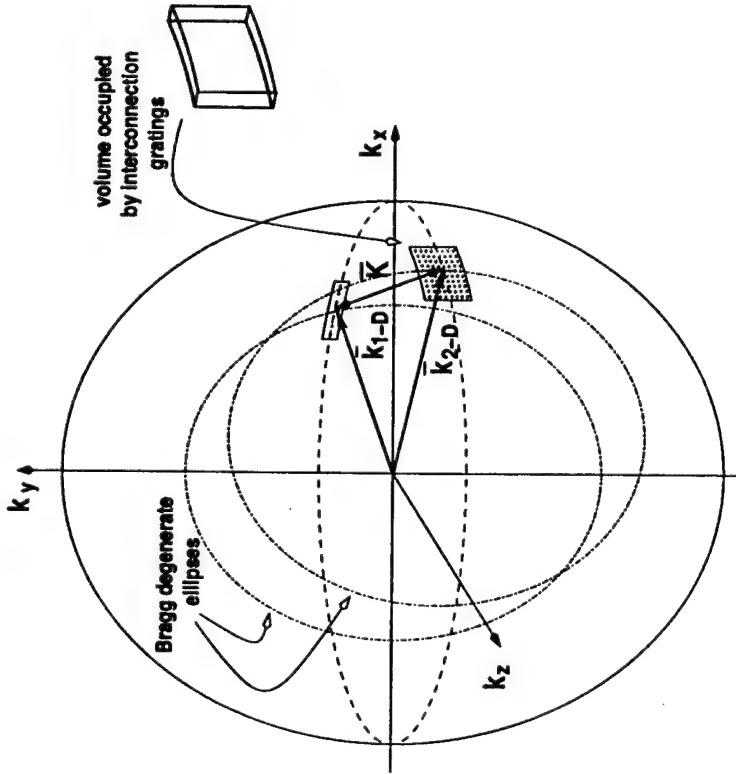
Figure 3.7: Volume holographic interconnections between a 2-D plane and a 1-D vector in k-space. The inset shows the volume of K-space occupied by all the interconnections.

crystal. For clarity, only the grating corresponding to the central wave vectors is drawn. Note that the central grating $\overline{K}$ is represented by a double tipped arrow. This is done in order to illustrate the fact that either the input vector or the input plane can be used for readout. In the input delay plane architecture, for example, readout is performed by the input plane. The Bragg condition determines that the diffracted beams will be directed towards the output vector (and towards the Bragg degenerate directions, to be explained shortly), this way interconnecting the input delay plane towards a vector of output detectors, as previously illustrated in figure 3.1. In the output delay plane
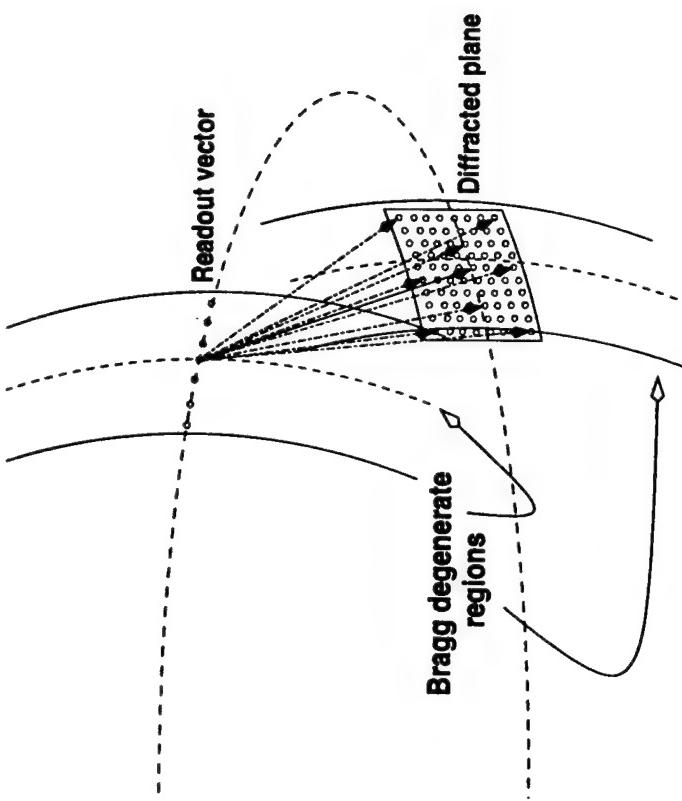
Figure 3.8: Readout vector and diffracted output plane in the output delay plane architecture.

architecture readout is performed by the input vector. Bragg matching determines that the readout beam will be directed towards the output detector plane, as illustrated in figure 3.5.

Figure 3.7 also indicates the position of the central Bragg degenerate ellipses, located in the wave vector surface. The total Bragg degenerate region is determined by the overlap of multiple degenerate ellipses, and it is shown in figure 3.9. Multiple diffraction orders occur if more than one readout wave vector is present along a degenerate ellipse. Note that in the case of readout by the input vector (as performed in the output delay plane architecture), only a single wave vector is present in the degenerate ellipses for any given readout pixel (Fig. 3.8). Therefore, the diffracted wave vectors point towards

Degenerate diffractions

Desired diffraction

Readout plane

Bragg degenerate regions
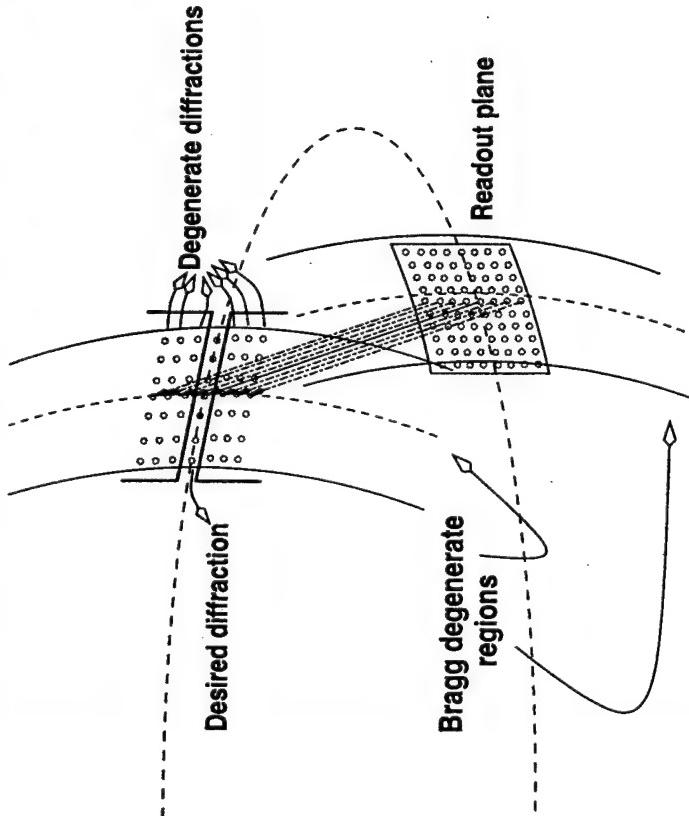
Figure 3.9: Bragg degenerate regions and degenerate diffraction orders during readout in the input delay plane architecture.

each pixel of the output plane and no Bragg degenerate diffraction occurs. In the case of readout by the input plane (as performed in the input delay plane architecture), pixel rows above and below the center row overlap with the degenerate ellipse, resulting in multiple diffracted beams above and below the output detector array. The Bragg degenerate beams can be spatially filtered by a slit centered on the detector array, or they can be simply ignored if they are not incident onto the area of the detectors or their circuitry.

3.3.2 Bipolar signal representation

The implementation of FIRNNs requires a method for correctly representing bipolar signals. If not, it would not be possible to represent zero-mean input signals, the error feedback term would not be capable of representing both positive and negative feedback for reinforcing signal enhancing and signal canceling correlations (as performed in a correlation-cancellation loop). Different methods exist for the representation of bipolar optical signals, each with their particular advantages and disadvantages. Some of them are:

- **intensity modulation** — the signal is represented by a variation of the optical intensity about a bias reference point [e.g., $I(t) = I_0 s(t) + I_{dc}$]. Intensity levels below the bias represent negative signals, and intensity levels above the bias represent positive signals.

- **dual rail intensity** — two intensity channels represent the signal: one of them represents positive intensities while the other represents negative intensities. The actual signal level is given by their difference [e.g., if $s(t) > 0 \Rightarrow I_p(t) = I_0 s(t)$, else $\Rightarrow I_n(t) = -I_0 s(t)$. $I(t) = I_p(t) - I_n(t)$].

- **field modulation** — the signal is represented by a modulation of the optical electric field [e.g. $E(t) = E_0 e^{i\omega t} s(t) = E_0 e^{i\omega t + \phi_s(t)} |s(t)|$, where $\phi_s(t)$ is the phase of $s(t)$]. Interferometric methods are required for signal detection.

The intensity modulation method has the advantage of easy implementation. In fact, it was the method used in order to achieve the first results early in the experimental implementation. However, it has some severe limitations. The DC bias (I_{dc}) produces a strong correlation in the correlation-convolution operations, largely dominating the underlying signal. Also, the bias intensity causes a quicker overall erasure of the PR grating, decreasing its maximum dynamic range.

Figure 3.10: Single-layer FIRNN based on the input delay plane architecture. Solid lines represent optical signals and dashed lines represent electronic signals.

3.4.1 Input delay plane architecture

Figure 3.10 shows a single-layer architecture which is based on the conventional temporal back-propagation algorithm, making use of space integration in forward propagation. An experimental implementation of this architecture applied to the special case of a single linear output is known as the FIRONN (finite impulse response optical neural network) architecture, explained in detail in chapter 5. The scrolling SLM produces a spatio-temporal plane of delayed versions of the input signal $g(\kappa)$. The complex amplitude distribution due to a single SLM pixel (i, τ) modulated by signal $s_i(\kappa - \tau)$ can be written as

$$E_{i,\tau}^A(\kappa; y, z, t) = \eta_S E_0 \exp[i(\omega t - kz)] \, \text{rect}\left(\frac{z - i\Delta_z}{\Delta_z}\right) \text{rect}\left(\frac{y - \tau\Delta_y}{\Delta_y}\right)$$
$$\times \text{rect}\left(\frac{t - (\kappa - \tau)\Delta_t}{\Delta_t}\right) s_i(\kappa - \tau) \qquad (3.5)$$

where η_S is the modulation depth of the SLM, $\omega = 2\pi\nu_o$ is the optical angular frequency, $k = \omega/c$ is the optical momentum, $\text{rect}(\cdot)$ is the rectangular function, Δ_z and Δ_y are

Dual rail representation [83] consists of the next step in complexity, requiring the use of separate channels for the positive and negative intensities. This requirement translates into a factor of two in increase in the required number of modulators and detectors, and/or a factor of two decrease in the resolution available on the SLMs. Also, it does not solve some of the problems with pure intensity modulation, such as the DC bias which would still be present in both channels. Moreover, new issues have to be dealt with, such as what to do when both channels are close to saturation, but the signal they represent (their difference) is close to zero. Thus, the added complexity required by this method does not justify its advantages when applied to the implementation of FIRNNs.

Field modulation, on the other hand, requires interferometric stability during heterodyne detection. This requirement usually translates into the need of controlled feedback of the optical path in order to maintain long term stability. However, if the signals represented have no DC component (and, for most applications, they either do not have any, or can be made not to have any), field modulation provides us with a zero-DC signal. Heterodyne detection, on the other hand, still provides us with a DC term [i.e., $v(t) \propto |E_r|^2 + |E_0|^2 + E_r^* E_0 s(t) + c.c.$], but it can be easily removed either by filtering, or by using differential detection, as explained in section 5.3.4. Thus, field modulation will be used in the following analyses and in the experimental implementation (Chp. 5).

3.4 Single-layer architectures

The input and output delay plane architectures can be extended into complete single-layer FIRNN architectures capable of gradient-descent on-line learning. The design for these architectures is presented and analyzed in this section.

the pixel pitch in the z and y directions, and Δ_t is the time interval between time-steps [to avoid confusion between the spatial index i and the imaginary number $i = \sqrt{-1}$, let us assume throughout this thesis that i always represents a spatial index, unless if it is inside an exponential [e.g. $\exp(i\omega t)$], when it is assumed to always represent $\sqrt{-1}$].

The total field modulated by the SLM is given by the sum over all the pixels

$$E^A(\kappa; y, z, t) = \sum_{i=1}^{N(0)} \sum_{\tau=0}^{T(0)} E_{i,\tau}^A(\kappa; y, z, t). \quad (3.6)$$

This spatio-temporal plane, modulated by the scrolling SLM, is propagated through a dynamic hologram, implemented by a PRC. At the crystal, the field from the SLM is diffracted by the volume grating representing the interconnection weights. The diffracted output is detected using a detector array that is in an exact image plane of the error modulator array, producing signal $\underline{\alpha}(\kappa)$ via spatial integration. Representing the instantaneous grating weights by $G_{ji,\tau}(\kappa)$, the voltage at the output detector j is given by

$$\sigma_j(\kappa) = Z_o \eta_d \left| \sum_{i,\tau} G_{ji,\tau}(\kappa) E_{i,\tau}^A(\kappa) + E_r \right|^2, \quad (3.7)$$

where η_d is the detector responsivity [A/W], Z_o is its output transimpedance gain [Ω], and E_r is the heterodyne reference. Note that, for simplicity of notation, it was assumed that the PRC is at the image plane of the SLM. Refer to appendix A for a more detailed operator-notation analysis of the holographic grating formation and its subsequent diffractive readout in which the crystal is placed in an arbitrary plane.

The non-linear operations are performed electronically, as depicted in the dotted balloon in figure 3.10, producing the final output $\underline{o}(\kappa) = f(\underline{a}(\kappa))$ and the delta error term $\underline{\delta}(\kappa) = g f'(\underline{a}(\kappa))[\underline{o}(\kappa) - \underline{d}(\kappa)]$, where g is an electronic gain that can be varied in order to control the learning rate (η). The term $\underline{\delta}$ is fed back electronically and is used to modulate a 1-D array of modulators which are precisely imaged through the volume hologram pixel matched onto the detectors. The modulated beam at the jth modulator

is given by

$$E_j^B(\kappa; x, y, t) = \eta_m E_0 g\left(\frac{y}{w_y}\right) \exp[i(\omega t - kz)] \, \text{rect}\left(\frac{x - j\Delta_x}{\Delta_x'}\right) \text{rect}\left(\frac{t - \kappa\Delta_t}{\Delta_t'}\right) \delta_j(\kappa), \quad (3.8)$$

where η_m is the modulator efficiency, $g(y) = \exp(-y^2)$ is a Gaussian distribution along the y dimension with a beam waist w_y, and Δ_x is the pixel pitch of the 1-D modulator.

The total modulated field is given by

$$E^B(\kappa; x, y, t) = \sum_{j=0}^{N(1)} E_j^B(\kappa; x, y, t). \quad (3.9)$$

The modulated beam interferes with the input plane $\underline{g}(\kappa)$ at the PR crystal. This volumetric interference pattern produces an outer-product between the input and error signals as the field modulated by each pixel on the SLM interferes with the fields of the error modulator array, modulated by every pixel of the error modulator. As the PR crystal time integrates this pattern, the grating accumulates the interconnections necessary to implement the adaptive weights, such that the instantaneous grating is given by

$$\dot{G}_{ji,\tau}(\kappa) = \beta E_j^B(\kappa) E_{i,\tau}^{A*}(\kappa) - \frac{G_{ji,\tau}(\kappa)}{\tau_{pr}}. \quad (3.10)$$

The second term in equation 3.10 accounts for the PR grating decay. It has the un-desirable effect of limiting the maximum diffraction efficiency of the weights being represented. On the other hand, it also functions as a regulatory decay, preventing the network from over-specializing during learning, allowing improved generalization performance if convergence can be achieved, which can be especially useful when the input signals are not WSS.

Note that equations 3.7 and 3.10 are the physical implementation of algorithmic equations 2.2 and 2.5. Also note that in this architecture, during each time-step Δ_t, $N^{(1)}$ operations are performed electronically (calculation of non-linear functions and error estimation) while $2N^{(0)}N^{(1)}T$ operations are performed optically (inner-product in forward propagation and outer-product in weight update) providing a cubic order

computational complexity of the operations being performed optically. Moreover, in the usual case when the network has less output nodes than input nodes, the number of electronic components and the number of error modulators are reduced. Since in this architecture the delay lines are present at the input (i.e. at the scrolling SLM), it is denoted as the input delay plane architecture.

3.4.2 Output delay plane architecture

Figure 3.11 shows a single layer FIRNN that uses the delayed feedback back-propagation algorithm, which makes use of time and space integration in forward propagation. In this case, the input vector is represented by an array of modulators which is imaged onto a volume hologram. The complex field amplitude of the modulated beam at the PRC can be expressed as

$$E_i^A(\kappa; y, z, t) = \eta_m E_0\, g\left(\frac{y}{w_y}\right) \exp\left[i(\omega t + kx)\right] \mathrm{rect}\left(\frac{z - i\Delta_z}{\Delta_z'}\right) \mathrm{rect}\left(\frac{t - \kappa\Delta_t}{\Delta_t'}\right) s_i(\kappa). \tag{3.11}$$

and the total modulated field is given by

$$E^A(\kappa; y, z, t) = \sum_{i=1}^{N^{(0)}} E_i^A(\kappa; y, z, t). \tag{3.12}$$

The modulated beam is diffracted by the PR gratings $G_{ji,\tau}$ representing the interconnection weights, and the diffracted beam is spatially integrated by a lens and detected by a TDI detector array. The signal at detector position (j, γ) between times $(\kappa - 1)\Delta_t$ and $\kappa\Delta_t$ is given by

$$a_{j,\gamma}(\kappa) = \eta_d \sum_{\tau=0}^{\gamma} \int_{(\kappa-1)\Delta_t}^{\kappa\Delta_t} \left| \sum_{i=1}^{N^{(1)}} G_{ji,\tau-\gamma+\tau}(\kappa + T - \gamma - \tau)\, E_{i,\tau}^A(\kappa) + E_r \right|^2\, dt. \tag{3.13}$$

At the output, when $\gamma = T^{(0)}$, assuming $\tau_{pm} \gg \Delta_t$, the output voltage of the interferometric term of interest is given by

$$\sigma_j(\kappa) \propto E_r\, a_{j,T}(\kappa) = E_r\, \eta_d \sum_{i,\tau} G_{ji,\tau}(\kappa - \tau)\, E_{i,\tau}^A(\kappa) + \mathrm{bias} \tag{3.14}$$

Figure 3.11: Single-layer FIRNN based on the output delay plane architecture. Solid lines represent optic signals and dashed lines represent electronic signals.

The output signal from the scrolling detector undergoes exactly the same electronic processing as in the input delay plane architecture [Fig 3.10] producing output signal $\underline{o}(\kappa) = f(\underline{a}(\kappa))$ and the error term $\underline{\delta}(\kappa) = f'(\underline{a}(\kappa))[\underline{o}(\kappa) - \underline{d}(\kappa)]$. However, it is necessary to produce delayed versions of $\underline{\delta}(\kappa)$ in order to perform weight adaptation. This operation is performed by a scrolling SLM. Note that the scrolling time of the T time-delays in the scrolling SLM must be the same as the scrolling time of the same T delays in the TDI-CCD (in other words, they have to shift at the same rate). In the scrolling SLM, the delayed versions of the error signal form the error plane $\underline{\delta}(\kappa)$ that is imaged onto the output TDI-CCD through the PRC. The complex amplitude field distribution of a pixel from this plane is described by

$$E_{j,\tau}^B(\kappa; x, y, t) = \eta_m E_0 \exp\left[i(\omega t - kz)\right] \mathrm{rect}\left(\frac{x - j\Delta_x}{\Delta_x}\right) \mathrm{rect}\left(\frac{y - \tau\Delta_y}{\Delta_y}\right)$$
$$\times \mathrm{rect}\left(\frac{t - (\kappa - \tau)\Delta_t}{\Delta_t}\right) \delta_j(\kappa - \tau), \tag{3.15}$$

and the total field is given by

$$E^B(\kappa; x, y, t) = \sum_{j=1}^{N^{(1)}} \sum_{\tau=0}^{T^{(0)}} E_{j,\tau}^B(\kappa; x, y, t). \tag{3.16}$$

Figure 3.12: Optical architecture based on the temporal back-propagation learning algorithm. Optic signals are represented by solid lines and electronic signals are represented by dashed lines. Operations inside the gray boxes are performed electronically. The input delay plane architecture is used for forward propagation and the output delay plane architecture is used for backward propagation of the δ terms.

similar to the one explained in Fig. 3.10 is used in forward propagation. The spatial-temporal plane $\underline{g}^{(0)}(\kappa)$ is produced by the scrolling SLM 1. This plane is diffracted by the adaptive weight gratings recorded in the volume hologram PRC 1. The array of diffracted signals, $\underline{g}^{(1)}(\kappa)$, is detected by the 1-D detector array 1. An electronic circuit performs the non-linear functions $f(\cdot)$ and $f'(\cdot)$, respectively, thereby producing the signals $\underline{s}^{(1)}(\kappa)$ and $f'(\underline{g}^{(1)}(\kappa-T))$, respectively. The columns of signal $\underline{s}^{(1)}(\kappa)$ are delayed in the second scrolling SLM, producing signal plane $\underline{\bar{s}}^{(1)}(\kappa)$ which is then diffracted by the weight gratings in the second volume hologram PRC 2. This diffraction produces the array of signals $\underline{g}^{(2)}(\kappa)$, which is detected by the 1-D detector array 2. This array generates the final network output $\underline{o}(\kappa)$. The voltage at output detector j is described by

$$\sigma_j^{(2)}(\kappa) = Z_0\,\eta_d \left| \sum_{i,\tau} G_{ji,\tau}^{(2)}(\kappa)\, E_{i,\tau}^A(\kappa) + E_r \right|^2, \qquad (3.18)$$

where $E_{i,\tau}^A(\kappa)$ is defined as in equations 3.5, but using $s_i^{(1)}(\kappa-\tau)$ as the modulation signal. Next, the desired signal $\underline{d}(\kappa)$ is electronically subtracted from the output to

The amplitude distribution $E_{j,\tau}^B(\kappa)$ interferes with a delayed version of the input signal $E_{i,\tau}^A(\kappa-T)$, and the resulting interference pattern modifies the PR grating according to

$$\dot{G}_{ji,\tau}(\kappa) = \beta\, B_{j,\tau}^B\, E_i^{A*}(\kappa-T) - \frac{G_{ji,\tau}(\kappa)}{\tau_{pr}}, \qquad (3.17)$$

thus providing us with the outer-product necessary to adapt the weights. Note that equations 3.14 and 3.17 implement equations 2.29 and 2.3, respectively. A simplified version of this architecture used in adaptive antenna array processing is known as the BEAMTAP architecture [74] in which a single output node is present and a single AOD is required. Note that once again, the non-linear operations with linear computational complexity are performed electronically, while the linear operations with cubic degree of computational complexity are performed optically. Since in this kind of architecture the time-delay-and-sum lines are present at the output (at the TDI-CCD), we denote it the output delay plane architecture.

3.5 Multi-layer architectures

Combination of the input delay plane and output delay plane architectures allows for the simultaneous implementation of forward propagation, backward error propagation, and weight updating in a multi-layer FIRNN. Once again, two kinds of architectures are possible: one based on the conventional algorithm and the other based on the delayed feedback back-propagation algorithm. For the sake of simplicity of notation, the following discussion will focus on 2-layer architectures, but the architectures presented can be extended to multiple layers with minimum effort. Let us first present the architecture based on the conventional algorithm, shown in figure 3.12.

3.5.1 Conventional temporal back-propagation

For the two-layer case depicted here, we have that $\Upsilon^n = 0$ for the last layer ($n = 0$), and $\Upsilon^n = T$ for the first layer ($n = 1$). An input delay plane architecture to

generate the weight update term $\underline{\delta}^{(2)}(\kappa)$ that is fed to the 1-D SLM 4. This plane is imaged through PRC 2 onto detector array 2. The second scrolling SLM displays $\overline{g}^{(1)}(\kappa)$. Interference of this signal array with the error gradient signal coming from SLM 4 produces the outer-product used for updating the weights in the volume hologram PRC 2. The grating evolution in PRC 2 is given by

$$\dot{G}_{ji,\tau}^{(2)}(\kappa) = \beta_2 E_j^B(\kappa) E_{i,\tau}^{A*}(\kappa) - \frac{G_{ji,\tau}(\kappa)}{\tau_{pr2}}, \quad (3.19)$$

where $E_j^B(\kappa)$ is defined in equation 3.8, but using $\delta_j^{(2)}(\kappa)$ as the modulation signal. In forward propagation, the grating at PRC 2 is read out by the beam coming from SLM 3 providing a diffracted beam in the direction of the beam coming from SLM 4 given by $G_{ji,\tau}^{(2)*}(\kappa) E_i^A(\kappa - \tau)$. Simultaneously, the output delay plane architecture is used to back-propagate the signal $\underline{\delta}^{(2)}(\kappa)$ modulated by the 1-D SLM 4 through the volume hologram PRC 2, providing a diffracted beam in the direction of the TDI-CCD [59] given by $G_{ji,\tau}^{(2)}(\kappa) E_j^B(\kappa)$. A detailed analysis of the adaptive grating formation and its readout by each of the writing beams is presented in appendix A. The total accumulated charges at each photon site of the TDI-CCD is given by

$$a_{i,\gamma}(\kappa) = \eta_d \sum_{\tau=0}^{\gamma}\int_{(\kappa-1)\Delta_t}^{\kappa\Delta_t}\left|\sum_{j=1}^{N^{(2)}} E_{j,\tau}^B(\kappa) G_{ji,\tau-\gamma+\tau}^{(2)*}(\kappa + T - \gamma - \tau) + E_\tau\right|^2 dt. \quad (3.20)$$

When $\gamma = T^{(1)}$, the output signal is given by

$$a_{i,T}(\kappa) = \eta_d E_\tau \sum_{j,\tau} E_{j,\tau}^B(\kappa) G_{ji,\tau}^{(2)*}(\kappa - \tau) + bias. \quad (3.21)$$

Thus, diffraction of the modulated term $\delta_j^{(2)}(\kappa)$ through the conjugate plane of the grating recorded in PRC 2 followed by space and time integration at the TDI-CCD provides us with a voltage signal which is proportional to the back-propagated error $\sum_{j,\tau} w_{ji,\tau}^{(1)} \delta_j^{(2)}$. This signal is electronically multiplied by the delayed derivative of the discriminant function, $f'(\underline{a}^{(1)}(\kappa - T))$, producing a signal proportional to the error gradient $\underline{\delta}^{(1)}(\kappa - T)$, which is applied to SLM 2. This signal is carefully imaged and

pixel aligned through PRC 1 onto detector array 1. The first layer requires the use of a delayed version of its input for updating the weights, since equation 2.16 in the 2-layer case becomes $\Delta w_{ji,\tau}^{(0)} = -\eta \delta_j^{(1)}(\kappa - T) s_i^{(0)}(\kappa - T - \tau)$. This can be achieved in two different ways: the first one consists of using a periodic input signal with a period of T time-steps, and the same signal can be used during forward propagation and weight update. The second method uses an SLM 1 with $2T$ rows which are dividing into two halves such that its first T rows are used in forward propagation, representing signal $\overline{g}^{(0)}(\kappa)$, and the remaining T rows represent signal $\overline{g}^{(0)}(\kappa - T)$. The two halves must then be optically combined before PRC 1. The two halves must be perfectly aligned, since the signal plane $\overline{g}^{(0)}(\kappa)$ reads the grating written by the signal plane $\overline{g}^{(0)}(\kappa - T)$ in combination with the error gradient $\underline{\delta}^{(1)}(\kappa - T)$.

This alignment can be performed by using a combiner, as shown in figure 3.13. A half wave-plate is used to flip the polarization in the delayed region $\underline{g}(\kappa - T)$ from ordinary into extraordinary polarization. The first birefringent crystal is cut at an angle α with respect to its uniaxial axis [Fig. 3.13(a)], causing the extraordinary beam to walk-off by an angle β [Fig. 3.13(b)] given by

$$\beta = \arctan\left[\frac{n_{e_1}^2(\alpha)}{2}\left(\frac{1}{n_{e_1}^2} - \frac{1}{n_{o_1}^2}\right)\sin 2\alpha\right], \quad (3.22)$$

where n_{e_1} and n_{o_1} are the extraordinary and ordinary indexes of refraction of the first crystal, and $n_{e_1}^2(\alpha)$ (the extraordinary index of refraction at an angle α with respect to the crystal $\overline{c}$-axis) is given by

$$n_{e_1}^2(\alpha) = \frac{n_{e_1}^2 n_{o_1}^2}{n_{e_1}^2 \cos^2\alpha + n_{o_1}^2 \sin^2\alpha}. \quad (3.23)$$

After propagation through the crystal length $L_1 = \frac{d}{\tan\beta}$, the walk-off superimposes the reading and writing half-planes of the delayed input plane. A second ninety-degree cut birefringent crystal is used to partially compensate for the optical path difference

3.5.2 Delayed-feedback back-propagation

The second multi-layer architecture (Fig. 3.14) makes use of the delayed feed-back back-propagation algorithm in order to implement a FIRNN. It is necessary to distinguish between the delays in the first and second layer, denoted $T^{(0)}$ and $T^{(1)}$, as defined in section 2. The input signal $\underline{s}^{(0)}(\kappa)$ is used to modulate a laser beam incident on a 1-D SLM. The modulated signal is diffracted by the volume hologram recorded in PRC 1 and spatially integrated at the individual photodetectors of the TDI-CCD 1. Here they are time integrated in a moving coordinate frame, as previously described in the output delay plane architecture. The output vector, $\underline{a}^{(1)}(\kappa)$, undergoes a non-linear transformation and produces the output of the first layer, $\underline{s}^{(1)}(\kappa)$. It is also used to calculate the derivative of the same non-linear function, $f'(\underline{a}^{(1)}(\kappa))$, which is time-delayed by $T^{(1)}$. Signal $\underline{s}^{(1)}(\kappa)$ is used to modulate another 1-D SLM, constituting the input to the second layer. Using the output delay plane architecture, the signal plane $\underline{\overline{s}}^{(1)}(\kappa)$ is diffracted by the hologram recorded in PRC 2 and the resulting product is spatio-temporally integrated at the TDI-CCD 2, producing the signal $\underline{a}^{(2)}(\kappa)$ which undergoes another non-linear transformation, producing the final output $\underline{o}(\kappa)$. The output is subtracted from the desired signal $\underline{d}(\kappa)$ and produces the error signal $\underline{e}(\kappa)$. This signal is multiplied by the derivative $f'(\underline{a}^{(2)}(\kappa))$, producing the term $\underline{\delta}^{(2)}(\kappa)$ used in back-propagation. Note that the scrolling SLMs and the TDI-CCDs are rotated by 90° in this architecture in comparison with the previous ones. This rotation, however, is not required for the correct system operation. It has been used in order to simplify the drawing of this architecture.

Weight update is first performed in the second layer, making use of the input delay plane architecture, where the error terms $\underline{\delta}^{(2)}(\kappa)$ are time delayed in the scrolling SLM 2 which is imaged onto the TDI-CCD 2 pixel by pixel. Note that the direction of scrolling of the TDI-CCDs are opposite to the direction of scrolling

Field Flattener · **Walkoff crystal** · **Diffraction compensation crystal**

(a)

Normal surfaces

Walk-off crystal · **Diffraction compensation crystal**

(b)

Figure 3.13: Optical combiner used to align the present version of the input plane with its delayed version. The first crystal is cut at an angle α with respect to the crystal axis $\overline{c}$, causing the extraordinarily polarized plane to walk-off towards the ordinarily polarized plane. The second crystal is used to produce equal radii of curvature. $\overline{c}$ – optical axis. $\overline{k}$ – optical momentum vector. $\overline{s}$ – Poynting vector.

between the ordinary and extraordinary beams, equalizing their wavefront curvature.

Its length L_2 is given by

$$L_2 = \frac{L_1}{n_{e_2} - n_{o_2}} \left[\frac{n_{o_1} - n_{e_1}(\alpha)}{\cos \beta} \right], \qquad (3.24)$$

where n_{e_2} and n_{o_2} are the extraordinary and ordinary indexes of refraction of the second crystal.

reached.

3.5.3 Read-write multiplexing

All of the architectures presented here are capable of forward propagation and weight update. These operations require either diffractive readout of the volume hologram (forward propagation) or its dynamic interferometric recording (weight update). In general the diffracted output beam is much weaker then the interfering beams used during holographic recording. Therefore, some form of isolation is required in order to prevent the output detectors from being overloaded with light from the writing beam imaged directly onto them. Different techniques for read-write isolation include wavelength multiplexing, polarization multiplexing [84], Bragg degenerate angle multiplexing [85], hybrid angle-polarization multiplexing in a uniaxial crystal [86], coherence multiplexing, frequency multiplexing and time multiplexing [41]. Whatever the technique employed, high read-write isolation is a stringent requirement for feedback learning applications, such as the detection of a weak signal in the presence of strong interference. High electronic gain can be added in order to amplify a weak detected signal, but the simultaneous amplification of the writing signal leaking through can cause the adaptive network to become unstable [87]. For a detailed report on an experimental implementation making use of temporal multiplexing, refer to chapter 5.

3.6 Summary

In this chapter, we realized that both FIRNN algorithms make use of computationally demanding spatio-temporal dot products in their implementation. Based on this fact, we explored two basic architectures capable of efficiently implementing the required dot products denoted the input delay plane and the output delay plane architectures. Both use a volume hologram for implementing the weighted interconnects. The first delays the input signal before performing the interconnections and the spatial


Figure 3.14: Optical architecture based on the delayed feedback back-propagation learning algorithm. Optic signals are represented by solid lines and electronic signals are represented by dashed lines. Operations inside the gray boxes are performed electronically. The output plane architecture is used for backward propagation of the output δ terms. The combiner used is the same described in the previous architecture.

SLMs. This is so because image inversion in a 4-f imaging system has been assumed. The resulting spatio-temporal plane $\overline{\underline{a}}^{(2)}(\kappa)$ interferes with a delayed version of the signal $\underline{g}^{(1)}(\kappa - T^{(1)})$, aligned by the combiner to propagate as a pixel matched beam collinear to $\overline{s}^{(1)}(\kappa)$, performing the outer-product necessary to alter the weights $\underline{\underline{w}}^{(1)}$ represented by the volume grating stored in PRC 2. The delayed error plane is also diffracted by the volume hologram and the diffracted beam is spatially integrated by a lens onto the detector array, which is a pixel matched image of the 1-D SLMs 2. The detected signal is then electronically multiplied by the delayed derivative $f'(\underline{a}(\kappa - T^{(1)}))$, producing the back-propagated term $\underline{\delta}^{(1)}(\kappa - T^{(1)})$, used to adapt the weights in the first layer. Again, the input delay plane architecture is used, with the spatio-temporal plane $\overline{\underline{a}}^{(1)}(\kappa - T^{(1)})$ imaged pixel by pixel onto the TDI-CCD 1 through the PRC 1, in which it interferes with a delayed version of the delayed input $\underline{g}^{(0)}(\kappa - T^{(0)} - T^{(1)})$, performing the outer-product necessary to alter the weights $\underline{\underline{w}}^{(0)}$ stored in PRC 1. In a 2-layer network we do not need the diffraction of $\overline{\underline{a}}^{(1)}$ off of the weights $\underline{\underline{w}}^{(0)}$, but in a multi-layer network we could keep backward propagating the error multiple times until the initial layer is

integration of the partial products, while the second performs the interconnections first followed by spatial integration and time integration in a TDI photodetector.

These two basic architectures were extended and provided with closed-loop feedback in order to implement FIRNN architectures capable of on-line gradient descent. Two single-layer architectures were presented: one based on the input delay plane architecture and the conventional temporal back-propagation algorithm, and the second based on the output delay plane architecture and the delayed-feedback back-propagation algorithm. Combining the two single-layer architectures it was possible to design multi-layer architectures capable of both forward and back-propagation with a continuous flow of information in the same set of weights. Again, two architectures were developed based on the two FIRNN algorithms. Both multi-layer architectures make use of similar optoelectronic devices with the main difference residing in the order in which the input delay plane or the output delay plane architectures are used. The conventional temporal back-propagation architecture presents the advantage that it does not require a delayed version of the input at the last layer, nor does it require the corresponding combiner. This advantage becomes less significant as the number of layers becomes larger. The delayed feedback architecture presents the advantage of requiring less tapped delay-lines (smaller TDI-CCDs and scrolling SLMs) when the number of nodes in the output is smaller than the number of nodes in the input.

Note that in all the architectures presented here, the cubic order operations (spatio-temporal convolution for forward propagation, spatio-temporal correlation for back-propagation and outer-product for weight update) have been performed optically, while the linear order non-linear operations have been performed electronically, thus exploiting the potential advantages of each technology.

Finally, the 2-layer architectures presented could potentially be extended to an arbitrary number of layers, where special care should be taken in the amount of delay applied to the fixed delay lines. Note that in a multi-layer implementation one could

create hybrid architectures by freely combining layers implemented by the conventional temporal back-propagation architecture with other layers implemented by the delayed feedback architecture. This is possible since the electronic signals $\mathfrak{s}^l(\kappa)$ and $\vec{\delta}^{l+1}(\kappa - \Upsilon^{L-l-1})$, the only ones passed between layers l and $l + 1$, are functionally identical, no matter which architecture is being used.

Chapter 4

Simulation and Analysis of Adaptive Array Processors

4.1 Introduction

This chapter presents simulations of different single-layer FIRNN architectures used as adaptive antenna arrays. Adaptive antenna array processors consist of systems capable of adaptively combining the signals from multiple antenna elements in order to implement a required task, which usually consist of pointing the antenna pattern in a given direction (or multiple directions) while simultaneously pointing its nulls towards interferers, jammers, or particular noise sources [42]. The simulations here presented provide a flexible tool with which one can easily experiment with different input signal scenarios (*e.g.*, jammers and signals at different AOAs, bandwidths, power ratios, etc.), array parameters (size and shape of the array, number of time-delays, inter-element spacing, etc.), various details pertaining to the optoelectronic implementation (PR grating decay, modulator non-linearities, detector noise, etc.), and that allows the user to display quantities which would otherwise be difficult to observe in a physical implementation (*e.g.*, the array receptivity pattern, SINR progression, learning curve, etc.). Moreover, the simulations provide an additional tool for validating the experimental results.

The software used in the generation of all of the simulations presented in this chapter is explained in detail in appendix B. This chapter is divided in two main sections: Section 4.2 describes simulations conducted using the BEAMTAP version of the single-layer FIRNN algorithm, as depicted in figure 2.5. These simulations were the first to be conducted and were performed at a time when the actual optoelectronic implementation had not been fully defined. Thus, they concentrate on the algorithmic performance and adaptive array parameters, rather then the physical parameters of the actual implementation. Currently, an experimental implementation of this architecture using acousto-optic Bragg cells and TFDs is nearing completion [88, 89] . Section 4.3 describes simulations conducted using the conventional LMS version of the single-layer FIRNN algorithm, as depicted in figure 2.4. Note, however, that for a sufficiently small learning rate (when convergence is guaranteed), both algorithms converge to the Wiener solution, meaning that all the conclusions reached in section 4.2 are also valid for section 4.3. However, section 4.3 focuses on simulating some of the relevant experimental parameters, such as weight decay, input binary quantization, and noise, in order to determine which results should be expected from the experimental setup presented in chapter 5. As will be shown, the simulation results accurately predict the performance of the experimental results, validating our experiments and showing that a correct selection of the experimental parameters to be simulated was made.

The simulations presented in section 4.2 were conducted with times scaled to RF frequencies, in the GHz range, while the simulations presented in section 4.3 were scaled at sonar frequencies, in the kHz range. For most scenarios, however, the main figures of merit are the fractional bandwidth of the signals involved, their highest frequency with respect to the Nyquist frequency, and the number of taps in the delay-lines. Thus, the frequency multipliers (GHz or kHz) were used in order to give physical meaning to the specific scenario being simulated, and are irrelevant to the actual outcome of the simulation.

RF input to phased array
TFD
$o(\kappa)$
$d(\kappa)$
Array input
Error feedback AOD
$e(\kappa)$
Time
Space
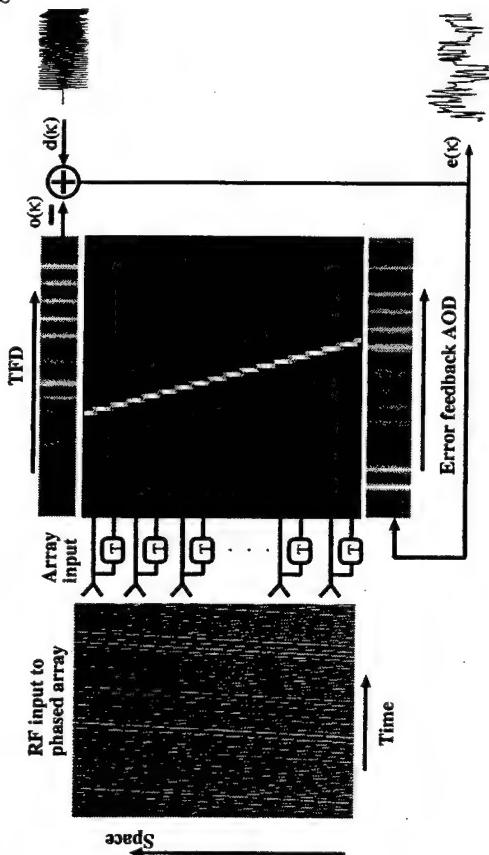
Figure 4.1: Diagram illustrating the simulation of the BEAMTAP architecture. The input signal is represented by the spatio-temporal signal on the left. At every time-step an instantaneous slice of the input is detected by the antenna array and it is mapped by the weight matrix (center of figure) to the output delay-line (top of the figure) by a vector-matrix multiplication. The output $o(\kappa)$ is produced by the time-delay-and-sum of the partial products conducted at the tap-in delay-line. The output signal is subtracted from the desired signal $d(\kappa)$, generating the feedback error signal $e(\kappa)$, which is fed through the error feedback tap-out delay-line (bottom of the figure). The weights are modified by the outer-product between the delayed error vector and the input vector delayed by T.

4.2 BEAMTAP simulations

Let us start our analysis by considering the BEAMTAP version of the single-layer FIRNN. Its optoelectronic architecture is depicted in figure 3.11 in the case of a single output channel. This system is capable of efficiently implementing a true-time-delay antenna array processor requiring the use of only two tapped delay-lines. Figure 4.1 illustrates the simulation of this architecture by showing some of the signals captured during a snapshot of the simulation during the processing of the signals in a 64-element array with 64 taps in its two delay-lines. The leftmost graphic depicts the spatio-temporal RF field amplitude sampled at 8 GSamples/sec along the horizontal time
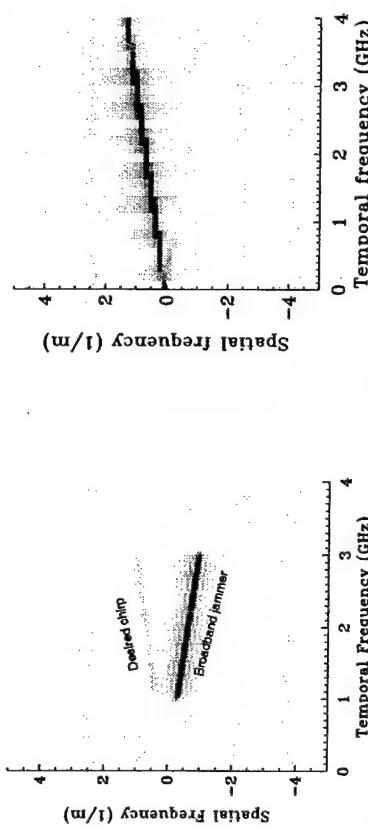
Figure 4.2: Spatial and temporal frequency spectrum of the input signal. The desired signal (a broadband chirp) is 100 times weaker than the broadband jammer.

Figure 4.3: Spatial and frequency spectrum of the weights after adaptation, showing a strong array response at the angle of arrival and bandwidth of the desired signal.

axis. This simulation illustrates the use of the BEAMTAP system in a communication scenario, in which the incoming RF signal to be detected consists of the desired signal $d(\kappa)$ added to the information signal $i(\kappa)$, with power 20 dB lower than the power of the desired signal. Furthermore, a broadband jammer is present with power 20 dB higher than that of the desired signal. Figure 4.2 shows these signals in the Fourier domain. The desired and information signal overlap in spectrum at an AOA of 0.2 radians. The broadband jammer ranges over the entire bandwidth of the desired signal, at an AOA of -0.2 radians. This combination of input signals is detected by each antenna element and is locally multiplied by the adaptive weights (center of Fig. 4.1). The partial products are temporally delayed and accumulated within the output delay-line (top of Fig. 4.1), producing an output signal $o(\kappa)$ which is subtracted from the desired signal $d(\kappa)$, producing the feedback error signal $e(\kappa)$. This signal is fed into the tap-out delay-line and its outer product with a delayed version of the input $[s(\kappa-T)]$ is used to adapt the weights. At steady state, the weights accumulate the cross-correlation of the
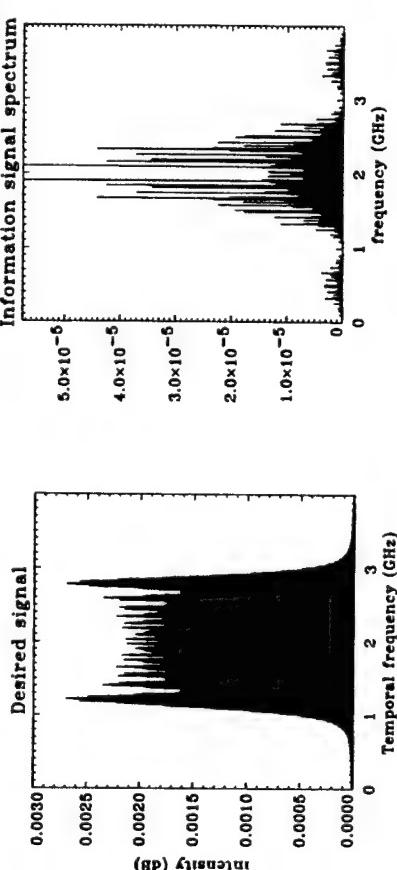
Figure 4.4: Spectrum of the desired signal, used to steer the array receptivity pattern toward the desired source.
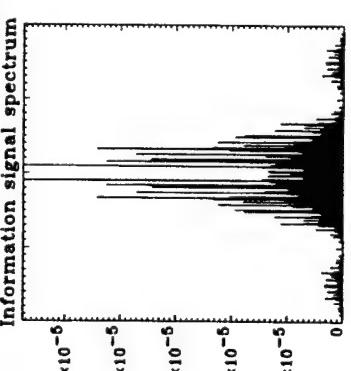
Figure 4.5: Spectrum of information bearing pseudo-random sequence added to the desired signal. Both signals overlap in spectrum.

Figure 4.6: Receptivity pattern after adaptation, showing a deep null at the jammer AOA.
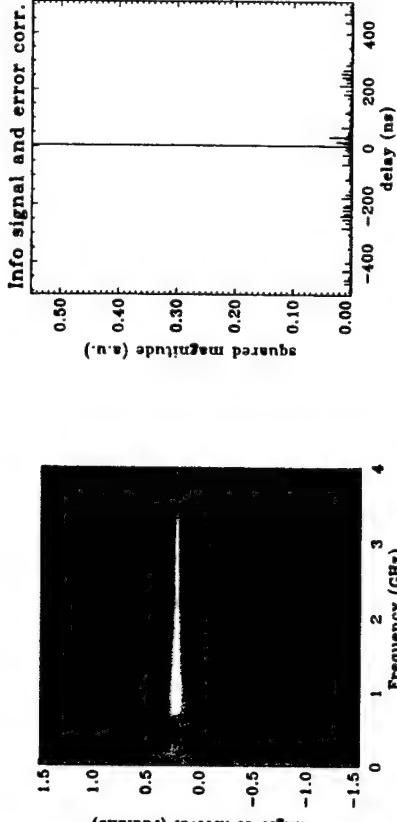
Figure 4.7: Cross-correlation between error feedback signal and pseudo-random sequence, showing successful signal detection.

desired signal with the input signal, producing the central band with sinc-like side-lobes seen in the center of figure 4.1. Figure 4.3 shows the final weights in the Fourier domain, showing a strong array response for signals with the same bandwidth and AOA of the desired source shown in figure 4.2.

Figure 4.4 depicts the spectrum of the broadband desired signal, which functions as a pilot steering signal, redirecting the antenna receptivity pattern towards its source upon adaptation. After adaptation, the weights project the spatio-temporal input signal towards the direction of the desired signal such that the feedback error signal becomes orthogonal to the desired signal. Consequently, uncorrelated information which is superimposed in the desired signal, even if at low power, will stand out in the feedback error signal. Figure 4.5 depicts the spectrum of an unknown PNBPSK (pseudo-noise bipolar phase shift keying) signal, which is first generated by a pseudo-random number sequence (making it spread-spectrum), and then modulated to the same center frequency and with the same bandwidth as the desired signal in order to produce an overlap of their

frequency spectra. This weak signal is added to the known pilot signal, generated by a far-field source, and it constitutes the information being transmitted.

Because the power of the information signal can be kept low, adaptation is not hindered. Figure 4.6 demonstrates this by showing the final receptivity pattern after adaptation in the presence of the information signal and a broadband jammer. The receptivity pattern shows the array frequency response at different angles-of-arrival and, therefore, it constitutes a powerful tool in the estimation of the adaptive array capability for beam-forming and jammer nulling. Note the strong response at 0.2 radians — the AOA of the desired signal — while the jammer is strongly nulled at -0.2 radians. The information signal had almost no affect in the final array receptivity pattern. Finally, figure 4.7 shows a plot of the cross-correlation between the feedback error signal and the pseudo-random sequence. The strong correlation peak proves that the pseudo-random sequence is present in the error signal and allows for various spread-spectrum communication techniques to be used in order to convey unknown information to the antenna array.

4.2.1 The SINR as a measurement of jammer nulling

In the previous example broadband jammer nulling was briefly introduced, but without a formal definition. An important performance parameter of an adaptive antenna array is its capacity to null jammers while still detecting as much of the desired signal as possible. Given the importance of this parameter, it is important to precisely define it. The definition must be able to compare the array performance before and after a jammer has been added. If not, one could select a jammer that already points towards a null in the receptivity pattern of the desired signal. In that case, no adaptation would be necessary, and the jammer would cause little interference to the signal detection and the adaptive array would erroneously be considered a good jammer nuller. Also, the definition must take in consideration how much of the desired signal is detected while the jammer is nulled. If not, one could set all the weights to zero. All jammers would be completely nulled, but so would the desired signal.

Let the input signal to noise plus interference ratio (SINR) be defined as the ratio between the power of the signal of interest P_s over the power of the unwanted interference plus noise $P_{ni} = P_i + P_n$. That is,

$$SINR = \frac{P_s}{P_{ni}}.$$ (4.1)

In order to make our definition of jammer nulling less dependent on the signal environment, let us define the output SINR (OSINR) as the ratio of the power of the system output when a unit power signal is used as the input, over the power of the system output when a unit power noise plus interference signal is used as the input, *i.e.*,

$$OSINR = \frac{E[|o(s)|^2]}{E[|o(i+n)|^2]},$$ (4.2)

where s is a unit power signal and $i + n$ are a unit power interference plus noise, and $o(\cdot)$ is the system output. Then we can define the jammer nulling depth J as the ratio

between the OSINRs before and after adapting to the jammers. That is,

$$J = \frac{OSINR_0}{OSINR_f},$$ (4.3)

where $OSINR_0$ is the final OSINR, after beam-forming towards the desired signal only, and $OSINR_f$ is the final OSINR after adapting in the presence of the desired signal, jammers, and noise. This definition applies equally well to narrowband and/or broadband signals, in the near-field and/or far-field, for conformal arrays, and for a different number of antenna elements and taps in the delay-line. Unfortunately, this parameter is difficult to measure experimentally, since in an optoelectronic implementation it is usually impossible to probe the array only with the desired or jamming signals without affecting the adaptive weights. However, when simulating an adaptive array it is possible to freeze the weights, probe the array, and proceed the simulation, thereby measuring the dynamics of jammer nulling and the relationship between jammer nulling and other array parameters, such as number of taps and the number of jammers.

4.2.2 Jammer nulling dynamics

Since a gradient descent algorithm is being used, we have that at every time-step the weights are adapted according to the negative gradient of the MSE, *i.e.*, $\frac{\partial w}{\partial t} = -\eta \frac{\partial \mathcal{E}}{\partial w}$.

From equation 1.6, we have that

$$\frac{\partial w}{\partial t} = -2\eta(Rw - p) \Rightarrow \frac{\partial w}{\partial t} + \eta'Rw = \eta'p,$$ (4.4)

where $\eta' = 2\eta$. Equation 4.4 can be solved by applying a unitary coordinate rotation Q to w, providing a new weight matrix $v = Q^\dagger w$ [90]. Applying this rotation to equation 4.4, we find

$$\frac{\partial v}{\partial t} + \eta'(Q^\dagger RQ)v = \eta'Q^\dagger p.$$ (4.5)

If Q is chosen so that $Q^\dagger RQ$ is a diagonal matrix, we have that the diagonal elements will be the eigenvalues of R, and equation 4.5 can be easily solved for v. Reversing the

coordinate rotation ($\mathbf{w} = \mathbf{Q}\mathbf{v}$), the solution for $\mathbf{w}$ is given by a linear combination of the solutions for $\mathbf{v}$, i.e.,

$$\mathbf{w}(t) = \sum_{l=1}^{NT} C_l e^{-\eta \lambda_l t} + \mathbf{R}^{-1}\mathbf{p}, \qquad (4.6)$$

where C_l are constant vectors determined from the initial value of $\mathbf{w}$, and λ_l are the eigenvalues of $\mathbf{R}$. Note that since the auto-correlation matrix $\mathbf{R} = E[\mathbf{s}\mathbf{s}^T]$ is positive definite, the eigenvalues of $\mathbf{R}$ are all positive and, consequently, equation 4.6 converges to the Wiener solution $\mathbf{w}^o = \mathbf{R}^{-1}\mathbf{p}$ (Eqn. 1.7) for $t \gg \frac{1}{\eta}\lambda_l$. Furthermore, note that the eigenvalues of $\mathbf{R}$ dictate the dynamics of the weight convergence: uniform eigenvalues all equal to σ — such as obtained from white noise — provide for a smooth convergence towards the Wiener solution, while a large disparity in the eigenvalues — such as obtained from strong far-field CW jammers in the input — determine the existence of regions of fast convergence (dictated by the larger eigenvalues), followed by regions of slower convergence (dictated by the smaller eigenvalues).

The slices of the receptivity pattern are useful in studying the adaptive system behavior, since they show in detail the system's temporal frequency response at a given AOA at some point in the adaptation process. Figure 4.8(a) shows the corresponding slice at the beginning of adaptation in the presence of two continuous wave (CW) jammers at 0.8 GHz and 2.3 GHz. The AOAs are 0.25 radians for the desired signal, 0.4 radians for the first jammer and 0.1 radians for the second jammer. The first jammer is 10 dB stronger than the desired signal, while the second jammer is 30 dB stronger than the desired signal. Notice that the stronger jammer is nulled much faster than the weaker one. This is explained by the fact that the stronger jammer produces larger eigenvalues in the correlation matrix $\mathbf{R}$. Thus the gradient of the error surface has a stronger component in the direction of the corresponding eigenvector — and the weights, which follow a gradient descent algorithm, tend to be updated predominantly in order to first null the stronger jammer.
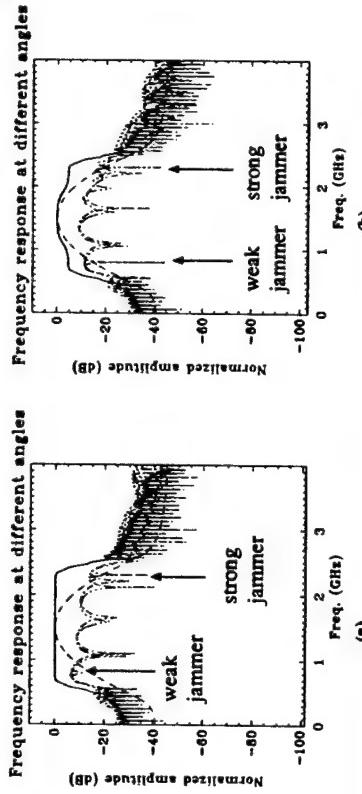
Figure 4.8: Array receptivity to two different jammers at different adaption times. (a) After 24576 iterations (6.14 μs). (b) After 8192000 iterations (1.02 ms). Solid line: array response at the AOA of the desired signal. Dashed line: spectrum of the desired signal. Dash-dotted lines: array response at the AOA of the jammers.

Figure 4.8(b) shows the same plot after the adaptation has reached steady-state. Note that at that point both jammers are almost equally nulled. This is because even though the weaker jammer is represented by smaller eigenvalues of the correlation matrix, after a sufficient number of iterations the contribution due to these eigenvalues equals the contribution due to the eigenvalues of the stronger jammer. Also note that the array's response at the AOA of the desired signal (solid line) has been adaptively reduced at higher and lower frequencies. This result is what should be expected from Wiener filtering, where the processor is capable of adaptively obtaining a better SINR by nulling some of its response at the AOA of the desired signal — an effect known as "in-band nulling" — in order to provide lower gain to the jammer side-lobes present at that angle of arrival.

The plot in figure 4.9 illustrates the same principle in a different manner. It shows the evolution of the output power due to the jammers, applied one at a time, in two separate adaptations: the first (solid line) using a strong jammer 30 dB above the desired signal, with an AOA of 0.1 radians, at 2.3 GHz (the same strong jammer

Output power comparison
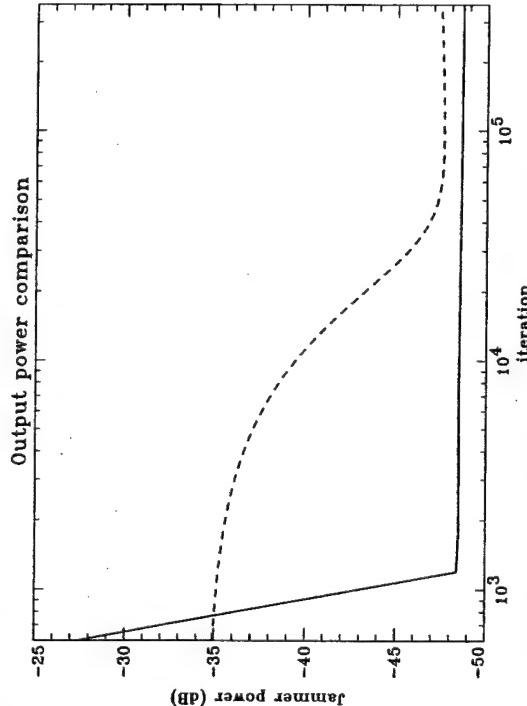
Figure 4.9: Jammer output power comparison between strong (solid line) and weak (dashed line) jammers during adaptation. The strong jammer (20 dB above the weak jammer) is nulled much faster.

used in the previous example). In the second adaptation (dashed line), the AOA and frequency of the jammer are kept the same, but its power is reduced to 10 dB above the signal power. The desired chirp has the same characteristics as the chirp used in the previous example (AOA = 0.25 radians, from 0.5 GHz to 2.5 GHz). In order to show the data in a log-log plot, both adaptations start from the same set of weights beam-formed towards the desired chirp alone (so that the initial output power is not zero and jammer nulling is minimally affected by beam-forming), and the first data point is taken after 512 iterations. Note that even though the stronger jammer initially has much more output power than the weaker version of the same jammer, it is nulled much more quickly, after about 1024 iterations, while the weaker jammer takes about 40 thousand iterations to reach an equivalent output power level.

4.2.3 Number of taps versus SINR

A conventional approach for selecting the minimum number of time-delay taps T necessary to avoid beam-squint in a time-delay-and-sum adaptive antenna array is to set it equal to the number of time-delays required to accommodate the maximum signal inverse bandwidth over the time-of-flight of the input signal across the array [6]. That is,

$$T = \frac{f_{max} L \sin \theta_{max}}{c}, \qquad (4.7)$$

where f_{max} is the highest frequency of the signal of interest, L is the array length, θ_{max} is the maximum AOA of all signals, and c is the speed of light. Such an approach guarantees that the peak of the cross-correlation between the input signal of interest and the desired signal can be fully contained within the weight matrix. It ignores, however, the possible contribution that the sidebands of the auto-correlation function might have on the performance of the array processor. The number of taps in the delay-lines of an adaptive array controls the number of degrees of freedom in the temporal dimension of that array. The higher the number of taps, the higher the order of the FIR filter corresponding to each antenna element and, therefore, the more complex can be the resulting frequency response. This allows for sharper edges at the cut-off bands, for example, allowing for better band-pass filtering and noise rejection. This also enables better adaptive shaping of the passing band, as exemplified by the in-band nulling shown in figure 4.8(b). However, it is also important to consider the interplay between the temporal and spatial dimensions. That is, even when the number of antenna elements is small a better SINR can be achieved by increasing the number of taps in the delay-lines.

The following example is especially devised in order to illustrate a case in which the conventional approach for selecting the number of time-delays underestimates the number of time-delays required for optimizing the SINR: figure 4.10 shows the signals

Figure 4.11: Fourier space representation of the input signal, showing considerable overlap in their spectra.

Figure 4.10: Two broadband jammers and a broadband chirp are poorly resolved by a 2-element antenna array.

Figure 4.12: Progression of the jammer and signal power present at the array output during adaptation of a 2-antenna array with 64 taps.

Figure 4.13: SINR progression during adaptation of a 2-antenna array with 64 taps.

detected by a 2-element array when the input consists of a broadband chirp from 1 GHz to 2 GHz at an AOA of 0.1 radians and two broadband jammers 10 dB stronger than the desired signal, the first ranging in frequency from 1 GHz to 1.5 GHz at an AOA of -0.1 radians, and the second ranging in frequency from 1.5 GHz to 2 GHz at an AOA of 0.3 radians. The spatial proximity of the signals and the small number of antenna elements make it very difficult to resolve the difference between these signals, even when looking at their Fourier spectrum, shown in figure 4.11.

However, by using a large number of taps, the processor becomes capable of correlating the input signal with a longer version of the desired signal, providing a larger processing gain for the antenna array, and making it possible to adaptively resolve the input signals. Figure 4.13 illustrates the dynamics of the signal detection and jammer nulling when using 64 taps in the delay-line, showing the evolution of the output power due to the desired signal (solid line) in comparison to the output power due to the jammers (dashed line), illustrating the fact that better performance is achieved by adaptively nulling the jammers while simultaneously improving signal detection. Note that it would not be possible to achieve similar performance by simply adding a long FIR filter at the output of the adaptive array instead of providing more taps to the delay-lines. The broadband jammers individually span the entire bandwidth of the desired signal and, therefore, no single FIR filter is capable of nulling them while simultaneously passing the desired signal. Figure 4.15 illustrates the SINR dynamical evolution during adaptation, given by the difference between the output powers due to the desired signal and jammers.

Figure 4.14 depicts the receptivity pattern after adaptation. Note that the array was not capable of producing a strong response at the AOA of the desired signal due to the lack of spatial degrees of freedom (i.e., spatial resolution). It was capable, however, of compensating for that effect by creating nulls that are sharp and deep in spectral frequency at the AOAs of the jammers thanks to the large number of temporal degrees
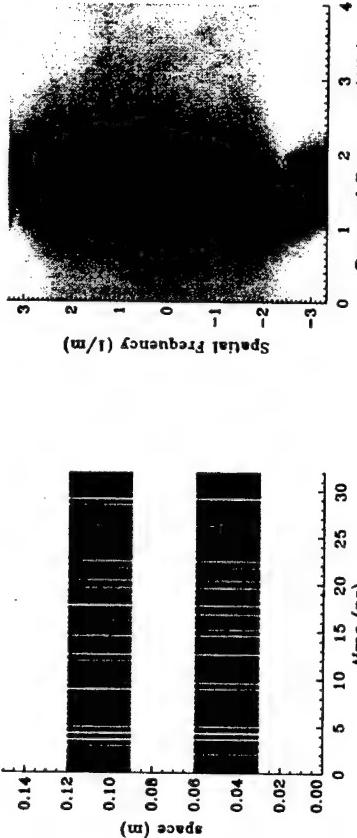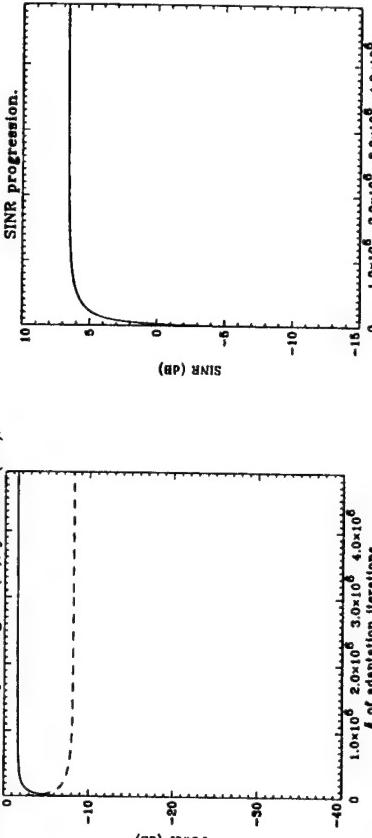
of freedom. This same scenario was repeated for arrays with varying number of taps, and the final SINR was recorded. A plot of the final SINR (at steady-state) as a function of the number of taps is shown in figure 4.15. More taps are always beneficial to the array performance, as expected, but diminishing returns would induce one to select a

bandwidth), incident on the array at random angles, as detected by the 64—element antenna array with 64 taps in the two delay-lines. Note that the multiple jammers cover a large area of the total spatio-temporal spectrum, an area which is proportional to the number of jammers present. The SINR evolution during adaptation (Fig. 4.17) is shown as a function of the number of jammers present in the spatio-temporal signal environment, where the number of jammers is varied from 1 to 256. The total jammer power and the learning rate are held constant as more jammers are added. Notice that the SINR always presents a quick initial improvement followed by a constant value at steady state, provided that the learning rate is kept within its convergence boundary. However, the rate of adaptation is seen to slow down in the presence of more jammers. This is due to the fact that when more jammers are present, the total jammer power is distributed more evenly amongst jammers and, consequently, no stronger jammer prevails, resulting in a small variation in the amplitude of the eigenvalues of the auto-correlation matrix **R**. Hence, the maximum boundary for convergence of the learning rate is reduced, and the weights tend to converge more slowly to the optimum solution.

Figure 4.18 shows a plot of the final SINR versus the number of jammers present, with the jammers kept at least 0.2 radians away from the AOA of the desired signal. Since the jammers are randomly placed in space, an average of 7 runs is used in order to present statistically significant results. Note that the SINR decays smoothly with the number of jammers, at a constant rate of about 1.12 dB per doubling of the number of jammers. In figure 4.19, the jammers were allowed to be randomly distributed over all angles, including the AOA of the desired signal. Consequently, the SINR decreased at a faster rate of 2.25 dB per doubling of the number of jammers, and when the jammer density reaches the critical value of 16, the SINR drops drastically at a rate of 13.33 dB per doubling of the number of jammers. It finally stabilizes at an SINR of about 4 dB. The sharp drop in the SINR happens because as the number of jammers

Figure 4.14: Receptivity pattern showing 2 nulls at the AOAs of the broadband jammers.

Figure 4.15: Array performance as a function of the number of taps in the delay-lines.

number of taps just after the knee of the curve (32 taps in this case).

Note that by selecting the number of taps to be equal to the value given by equation 4.7, one finds that $T = 0.15$, which greatly underestimates the necessary number of taps for best performance. That is to be expected, since a 2-element array can only point one beam and one null, whereas this scenario would require 2 nulls for best performance.

4.2.4 The Effect of Multiple Jammers on the System's Final SINR

The LMS algorithm, on which BEAMTAP is based, has the ability to null multiple simultaneous jammers, but the null depth may decrease with the number of jamming sources and their relative bandwidths. In addition, the nulling dynamics may also depend on the number of jammers and their bandwidth. The following simulations were developed to test and demonstrate these effects.

Figure 4.16 shows the spatial-temporal spectrum of a broadband desired chirp at broadside in the presence of 16 broadband jammers (all covering the same 1 to 4 GHz

Figure 4.19: Final SINR versus number of jammers. Jammers are allowed to overlap with the AOA of the desired signal.

Figure 4.18: Final SINR versus number of jammers. Jammers are kept 0.2 radians away from the AOA of the desired signal.

Figure 4.16: Fourier domain representation of sixteen randomly placed broadband jammers and a broadband desired signal (center).
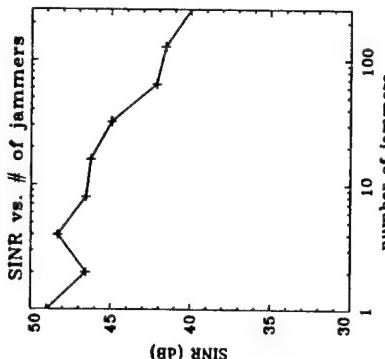
Figure 4.17: SINR progression during adaptation for an exponentially increasing number of jammers.

increases, the probability of randomly positioned jammers overlapping with the desired signal (e.g. main-beam jammers) also increases. This drop-off point is proportional to the number of antenna elements in the system, since more antenna elements allow the antenna directivity pattern to have a narrower main lobe corresponding to a higher spatial resolution, increasing the system's angular discrimination between signals at different AOAs.

We also studied the relationship between the number of jammers present and the adaptation time required to reach the steady state value for the SINR. Initially, it was expected that more time would be required to reach steady state as more jammers caused a reduction in the dynamic range of the amplitudes of the eigenvalues governing the adaptation, thus reducing the rate of convergence. Such an effect is observed. However, this effect is accompanied by a reduction in the final SINR, which thereby decreases the number of iterations necessary to reach steady state. However, if the learning rate had been increased as the reduction in the maximum amplitude of the eigenvalues allowed

it to happen (while maintaining convergence), we would have noticed a reduction in the time to convergence.

4.2.5 Input non-linearities in the spatio-temporal frequency domain

From its early architectures [91, 92] the BEAMTAP processors have been designed using electro-optic (EO) modulators in order to convert the RF input signals into RF-modulated optical signals. Inevitably, all such optical intensity modulators suffer from non-linear distortions that reduce the dynamic range of the system by producing spurious inter-modulation products that can produce spectral peaks above the noise floor. Figure 4.20 shows a schematic diagram of such a modulator and its non-linear response. The diagram shows an integrated Mach-Zender interferometric modulator, which operates by applying an input voltage V_{in} to one of the arms of the interferometer. The electro-optic effect [93] causes a change of the phase of the beam going through this arm which is added to the unmodulated beam ($E_0 \exp(i\omega t) + c.c.$), providing the output field

$$E_0 \exp(i\omega t) \left[1 + \exp\left(i\frac{\pi}{V_\pi} V_{in}(t) \right) \right] + c.c. \text{ When a photodetector in a direct optical link is}$$

(a)

(b)
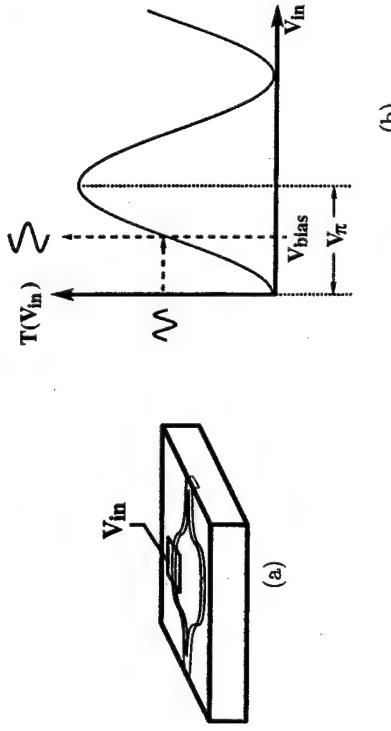
Figure 4.20: Schematic representation of a Mach-Zender EO interferometer modulator (a) and its non-linear response (b). The input voltage (V_{in}) is biased by $V_{bias} = \frac{V_\pi}{2}$, the input voltage required to provide an output with minimum even orders of harmonic distortion.

illuminated with light coming from such an EO modulator, the resulting photocurrent is given by [94]

$$i(t) = \frac{I(t)\mathcal{R}}{2}\left\{1 - \cos\left[\Gamma_0 + \frac{\pi}{V_\pi}V_{in}(t)\right]\right\}, \qquad (4.8)$$

where $I(t)$ is the optical input power of the modulator, $\mathcal{R}$ is the responsivity of the photodetector, Γ_0 is the retardation at the bias point, and V_π is the half-wave voltage. These modulators are commonly used at a bias point V_{bias} located at one half V_π, in order to prevent even orders of harmonic distortion. This provides the modulator with a raised sinusoidal modulation pattern which, due to its odd symmetry, gives rise to a the odd orders of non-linearity, and larger applied voltage swings result in higher amounts of non-linearity present at the modulated signal. Moreover, if the bias point is not set exactly at the mid-point of the sinusoidal, even orders of non-linearity also occur.

When a temporal signal with frequencies components f_1 and f_2 undergoes non-linear modulation, the even-ordered non-linear terms give rise to an output signal of frequency components $2f_1$, $2f_2$, $4f_1$, $4f_2$, $f_1 - f_2$, $f_2 - f_1$, $f_1 + f_2$, ... The odd-ordered terms, on the other hand, give rise to the frequency components $3f_1$, $3f_2$, $5f_1$, $5f_2$,

$2f_2 - f_1$, $2f_1 - f_2$, $2f_2 + f_1$, $2f_1 + f_2$, ... In our antenna array simulations we have studied the effect of different values of V_π and random variations of the bias point with respect to each individual antenna element. In an array we must consider both the spatial and temporal frequencies when evaluating the effects of non-linear inter-modulation. This must be done for the even-ordered as well as odd-ordered inter-modulations, since it may be impractical to individually adjust the bias point of the N modulators in the input in order to eliminate the even-ordered inter-modulations.

Figure 4.21 illustrates in the spatio-temporal frequency plane the effect of input non-linearities in a single far-field CW tone using double side band (DSB) modulation over an optical carrier of frequency ν_0. The diagram on the top of the figure shows the positive and negative conjugates of the DSB modulated signal over the optical carrier. The inset at the bottom of the figure zooms into the positive conjugate, showing in detail the modulated signal and its non-linear terms. A CW signal has a single temporal frequency and a single spatial frequency and, therefore, is represented by a single point and its complex conjugate in the spatio-temporal plane. Input non-linearities, however, give rise to higher order terms, here shown up to the third order. These terms can be found in the spatio-temporal frequency plane first by drawing a vector from the origin (DC) to the point representing the CW tone, and then performing multiple vector additions, as indicated by the dashed vectors in figure 4.21. Consequently, they all lie in the same tilted line and thus appear to be frequency components coming from the same AOA as the original signal. However, the LMS adaptive signal array is capable of adaptively modifying its spatio-temporal response in order to filter out those non-linear terms, as will be shown in section 4.2.6. Note that usually only the positive temporal frequencies are shown in the simulations, but their negative conjugates are always present since all simulated signals are real.

Figures 4.22 and 4.23 illustrate this principle through a simulation. The first figure (Fig. 4.22) shows a broadband chirp detected in the far-field with an AOA of

Figure 4.22: A broadband input signal in spatio-temporal frequency space, before non-linear modulation.

Figure 4.23: A broadband input signal after non-linear modulation, giving rise to a DC term and higher order non-linear terms.

the frequency components of the signal are doubled but, for Gaussian spectra, the rate of decay from the central frequency also doubles. The lowest 3-dB frequency (1.5 GHz) gets translated to 3 GHz, the highest 3-dB frequency (2.5 GHz) gets translated to 5 GHz, and so on, but these components are now 6 dB below the peak, and the 3-dB points are now located between 2.59 GHz and 5.41 GHz. The third order non-linear term is seen centered at 6 GHz with 2 GHz of bandwidth. Finally, the fourth order can just barely be seen at 8 GHz with 2.83 GHz of bandwidth ($2\sqrt{2}$ GHz). Note that all the non-linear terms lie on the same straight line with respect to the original broadband signal, as shown in the previous example (Fig. 4.21). Note that a strong DC component is also present (Fig. 4.23), due to the bias of the raised cosine modulation.

What happens if two tones with different AOAs are detected? Figure 4.24 illustrates this case, where two CW tones (tone 1 and tone 2) are detected in the far-field. As in the previous case, statistical second order and deterministic third order non-linearities are present, represented by the dashed vectors. However, the mixing of the two tones also gives rise to 2nd order (dash-dot vectors) and 3rd order (dash-dot-dot vectors)

Figure 4.21: Top: an optical carrier of frequency ν_0 is DSB modulated by a CW tone. Bottom: inset showing in detail the CW signal and its 2nd and 3rd order non-linear terms.

0.12 radians, in a logarithmic intensity scale, sampled at 24 GSamples/sec in order to adequately sample the higher order non-linear terms. The signal is centered at 2 GHz with 1 GHz bandwidth. Figure 4.23 illustrates what happens when this same signal undergoes strong non-linear modulation, with the input signal amplitude varying from 0 to V_π, and with V_{bias} also varying randomly from 0 to V_π across the array of 64 modulators, in order to illustrate strong second order non-linearities, which is clearly seen at 4 GHz as well as near 0 GHz. Note that the second order term has 1.4 GHz of bandwidth — the bandwidth of the original signal times $\sqrt{2}$. This is so because all of

Figure 4.26: Broadband chirp and jammers in the Fourier domain with non-linear modulation. A 2-tone 3rd order inter-modulation product between the jammers falls on the desired signal bandwidth.

Figure 4.25: Broadband chirp and jammers in the spatio-temporal Fourier domain with linear modulation. No non-linear inter-modulation products are present. Log intensity scaled is used.

Figures 4.25 and 4.26 show simulations illustrating this case. Figure 4.25 shows the input signal spectrum without input non-linearities. It consists of a desired broadband chirp and two CW jammers, each jammer 20 dB stronger than the desired signal, the first at 2.5 GHz, with an AOA of 0.25 radians, and the second at 3 GHz, with an AOA of 0.4 radians. The desired signal is a broadband chirp ranging from 1 GHz to 3 GHz, at an AOA of 0.04 radians, specifically chosen such that the 2-tone 3rd order inter-modulation of the jammers falls exactly on the spatio-temporal signal spectrum. Figure 4.26 shows the same spectrum when the input signal undergoes non-linear modulation with its amplitude varying from $-V_\pi/2$ to $+V_\pi/2$, and with V_{bias} varying from $+V_\pi/4$ to $-V_\pi/4$ from its second harmonic nulling optimum setting ($V_\pi/2$), giving rise to second and third order inter-modulation products between the jammer and between the desired signal and the jammers. Also note that the spatio-temporal position of the two jammers is specifically chosen so that their third order inter-modulation falls on the

Figure 4.24: Spatio-temporal frequency representation of the detection of two CW tones with different frequencies and AOAs and their non-linear higher order terms (dashed lines), 2nd order 2-tone inter-modulation terms (dash-dot) and 3rd order 2-tone inter-modulations terms (dash-dot-dot). The dashed circles highlight non-linear terms which give rise to in-band interference.

inter-modulation terms. The four 2nd order inter-modulation terms are found by a vector addition between the vectors representing tone 1, tone 2, and their conjugates. The eight 3rd order inter-modulation terms can be found either by two vector additions between the vectors representing tones 1 and 2 and their conjugates, or simply by a single vector addition between the 2nd order terms and tones 1 and 2 and their conjugates. Note that, in the simulations, if the higher order terms are not Nyquist sampled (i.e, with a sampling frequency at least twice as large as their frequencies) aliasing will occur, resulting in a fold over of these terms back into the sampled spatio-temporal spectrum, but in a continuous time analog system as such, aliasing would occur.

spatio-temporal spectrum of the desired signal, acting as an in-band main-beam jammer. The next section discusses the effect that such non-linearities have on the array processor adaptation.

4.2.6 Adapting to non-linear inputs

This section presents the simulated results obtained when adaptation is performed using the non-linearly modulated inputs discussed in section 4.2.5, starting with beam-forming towards the broadband chirp (AOA = 0.12 radians, from 0.5 GHz to 2.5 GHz) shown in figures 4.22 and 4.23. First, adaptation is performed using a linear input signal, providing us with the smooth learning curve shown in figure 4.27 and the squint-free broadband receptivity pattern shown in figure 4.28. A strong untitled receptivity pattern is seen at 0.12 radians, the AOA of the desired signal, over its entire bandwidth. Grating lobes appear above 6 GHz, because the antenna spacing was chosen to be $\lambda/2$ at 6 GHz, but note that only noise arrive at these frequencies. Nevertheless, the adaptive algorithm attempts to null this out of band noise, so that a null is formed at the peak of the spatially aliased grating.

Next, adaptation is performed using the non-linearly modulated input, providing us with the smooth learning curve shown in figure 4.29, but with a value after 10^5 iterations (6.0×10^{-4}) not quite as low as in the previous case (3.6×10^{-4}). Since both

Figure 4.27: Learning curve under a linear broadband chirp input.
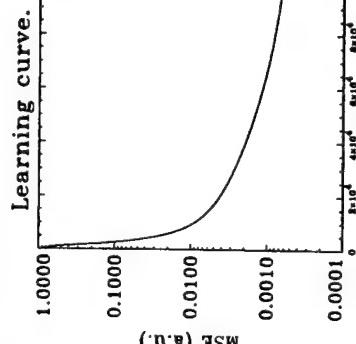
Figure 4.28: The receptivity pattern showing beam-forming towards a broadband chirp.
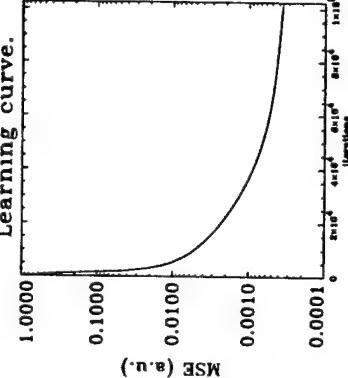
Figure 4.29: Learning curve, showing a well behaved adaptation under a non-linear broadband chirp input.
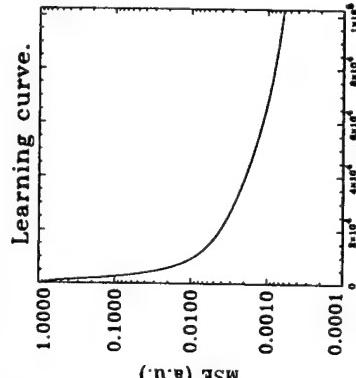
Figure 4.30: The receptivity pattern of the non-linearly modulated input signal, showing that the system has ignored the non-linear terms.

systems are still adapting the final asymptotic values could be similar, but the non-linear case is clearly a little slower both in the rapid initial phase as in the asymptotic convergence. A question arises as to whether or not the signal replications at higher frequencies will give rise to beam-formed signals at those frequencies as well. Fortuitously, the inter-modulations act as unwanted out-of-band main-beam interference sources, so that they are uncorrelated with the feedback error signal and, as such, the weights do not accumulate their correlations with the feedback signal, as shown in the receptivity pattern depicted in figure 4.30.

The next example consists of broadband beam-forming and jammer nulling in the presence of two CW jammers, depicted in figures 4.25 and 4.26. The frequencies and AOAs of the jammers were chosen so that the 2-tone third order inter-modulation between the positive frequency component of the first tone and the conjugate frequency

SINR progression.

Figure 4.33: SINR progression during adaptation under non-linear input modulation.

SINR progression.

Figure 4.32: SINR progression during adaptation to linear input modulation. Two CW jammers and a desired broadband chirp.

present at the AOA and frequency of the CW jammers. The final SINR obtained for this case is very high (74.3 dB), as shown in figure 4.32. Figure 4.33 shows the SINR progression plot obtained for the non-linear case, with a lower final value of 56.2 dB. Note that a larger number of iterations was necessary before reaching steady-state (1.6 million iterations, compared to 1.3 million iterations in the linear case). Figure 4.34 shows the receptivity pattern obtained when extremely strong non-linearities are present. Note that the processor attempts to adaptively null the in-band main beam jammer, but this is done in detriment to the array's capacity to detect the desired signal over its full bandwidth. Also note that in this scenario, since the signals and jammers are repetitive with a period of 8192 samples, they are coherent to each other as well as to the desired signal, an unlikely case which makes the nulling of the jammers even more difficult. Nevertheless, the processor was still capable of handling it to the best of its capacity, with acceptable results. Thus, one can conclude that a linear adaptive array processor is capable of operating in the presence of input non-linearities, but this compensation

Figure 4.31: Receptivity pattern for linear input modulation. The desired signal is strongly detected. Two deep nulls are adaptively placed over the position of the jammers, shown inside the dark circles.

component of the second tone falls exactly on the center of the spatio-temporal spectrum of the desired signal, effectively forming an in-band main beam jammer which is only present when the non-linearities are large. The desired signal consists of a broadband chirp ranging from 1 GHz to 3 GHz at an AOA of 0.04 radians. Figure 4.31 shows the receptivity pattern obtained without non-linearities, showing a strong response at the AOA and bandwidth of the desired signal. Two deep nulls (circled in black) are also

Figure 4.35: Diagram illustrating the simulation of the FIRONN architecture. Delayed versions of the input signal are represented by a scrolling SLM. PR gratings (center of figure) map the input plane to the output, where they are spatially integrated at the output photodetector. The output signal $o(\kappa)$ is subtracted from the desired signal $d(\kappa)$, generating the feedback error signal $e(\kappa)$. The weights are modified by the outer-product between the instantaneous error and the spatio-temporal input plane.

4.3   FIRONN simulations

This section presents simulations of the FIRONN architecture, as depicted in figure 3.11 for the case of a single output channel. This system uses a scrolling SLM for implementing the simultaneous delay of all input channels. Its optoelectronic implementation is analyzed in detail in chapter 5. Figure 4.35 illustrates the simulation of this architecture by showing some of the signals captured during a snapshot in time of the simulation during the adaptive processing of the input signals. Since the experimental implementation is well known, these simulations also take into account many of the physical parameters which are relevant to the system performance in an attempt to produce simulation results which accurately reproduce the system's performance. As will be shown, this goal has been achieved with remarkable results.

In the FIRONN architecture, a 256×256 electronically addressed binary FLC-SLM is driven by a Displaytech SLiM II driver, capable of storing up to 128 frames. These frames are displayed in succession in order to simulate the continuous arrival of data at a 256-element antenna array. At each time-step a new frame is displayed, each one displaced by two pixels with respect to the previous one, thereby simulating
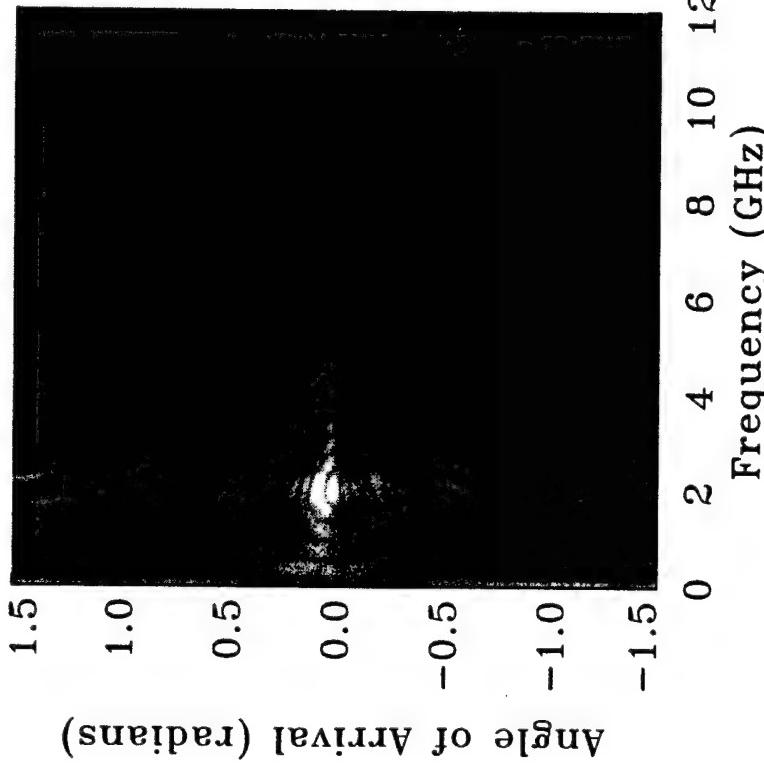
Figure 4.34: Receptivity pattern for non-linear input modulation. The response at the AOA of the desired signal is severely degraded.

can require the use of many of the available degrees of freedom — and will be followed by a degradation in the final output SINR, a degradation which is dependent on the particular input scenario.

256 delay-lines, each one with 128 taps. The signals represented are periodic with a period of 128 frames, so that it is possible to continuously represent the detection of the input signal. The simulations presented in this section faithfully reproduce the SLM resolution and the periodic repetition of the input signal.

As shown in figure 4.35, the time-delayed elements of the input plane produced by the SLM are multiplied by the individual elements of a weight matrix representing the volume holographic grating formed in a PR crystal (center of the figure). All the elements of the resulting product between the two matrices are summed by spatial integration with a lens, producing the instantaneous output $o(\kappa)$ on the output detector. The output signal is subtracted from a desired signal $d(\kappa)$, producing the feedback error signal $e(\kappa)$. Following the LMS learning rule, the weights are modified by the outer-product between the instantaneous error and the spatio-temporal input plane $\underline{g}(\kappa)$. At steady state, the weights $\underline{w}(\kappa)$ accumulate the cross-correlation of the desired signal with the input signal, producing the central band with sinc-like side-lobes seen in the center of figure 4.35.

4.3.1 Binary quantization of the input

In our experimental implementation, the use of FLC-SLMs requires the binary quantization of all of the input signals, which adds a hard-threshold non-linearity to the input and raises concerns regarding the rate of convergence and stability of the algorithm due to the large non-linear inter-modulations. The following simulations, therefore, have been designed in order to test the behavior of the adaptive array when the input signal is a broadband binary chirp in the far-field at an AOA of 0.25 radians in the presence of white noise 15 dB below the signal level, as depicted in figure 4.36. Figure 4.37 shows the spatio-temporal Fourier spectrum of the binary chirp input. Note that the non-linear binary quantization causes the appearance of higher order modulation terms which fold back in spectrum due to the spatial-temporal sampling of the input.
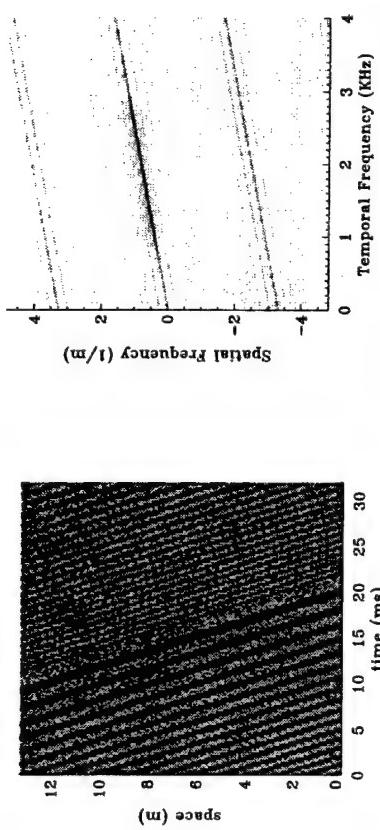
Figure 4.36: Far-field broadband chirp input with white noise, after binary quantization.

Figure 4.37: Spatio-temporal Fourier domain representation of the input. Aliased non-linear fold-over terms are present above and below the signal of interest.

Figure 4.38 shows the MSE during adaptation. Although 8 thousand iterations are required in order to bring the MSE down to 10^{-4} (at which point white noise dominates the feedback error signal), the LMS algorithm provides an approximate result even after the very first iteration. A matrix inversion algorithm, on the other hand, requires $(128 \times 256)^4 \approx 1.15 \times 10^{18}$ operations just to invert the $(128 \times 256)^2$ auto-correlation matrix $\mathbf{R}$ once, making its use prohibitive for large arrays. Also note the presence of two distinct convergence regions. This is because the desired signal is an analog chirp with a Gaussian profile. Thus, the system is capable of quickly detecting the AOA and bandwidth of the binary chirp. That is, the adaptive array requires less than 4 thousand iterations in order to point its directivity pattern in the direction of the desired signal, an operation usually denoted as **beam-forming**. However, the system takes a long time adapting its weights in order to adjust the frequency response of the detected signal into that of the desired signal. The receptivity pattern shown in figure 4.39 verifies that the AOA and bandwidth of the binary chirp are detected. Note that the non-linear terms
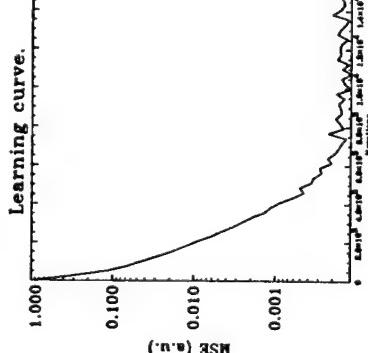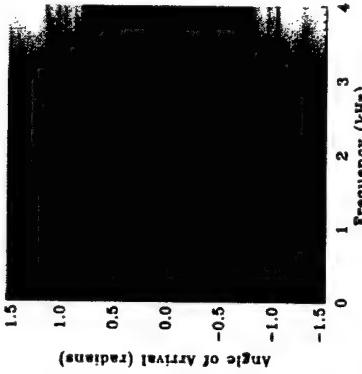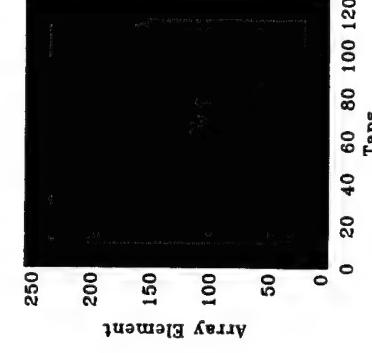
over into the spatio-temporal spectrum of the original signal (Fig. 4.41) and, since they are correlated with the feedback error signal, adaptively detects and linearly transforms them into the desired signal.

Note that quantization is not a commutative operation. In other words, quantizing individual signals and then summing them is different from quantizing their sum. In the following simulations, whenever more than one signal is present in the input (e.g. jammer nulling and beam-forming, multiple beam-forming, multiple jammer nulling, etc.) the quantization of the input array is always performed after the signals are combined. This is done in order to correctly simulate the detection of the combined input signals by an adaptive array, followed by binary quantization in the optoelectronic processor.

Figure 4.42 shows the desired binary chirp signal in time (top) and in frequency (bottom). Note that the signal ranges from a quarter of the Nyquist frequency f_N to 0.75 f_N. The desired signal used here is the same used in all of the simulations in section 4.3.

4.3.2 Exponential weight decay

Another concern regarding the experimental implementation is the effect of photorefractive grating decay with regards to the convergence of the adaptation and the overall performance of the adaptive array. It is known that a temporal decay term in the adaptive weights can operate as a regulatory term [95], preventing the network from over-specializing to a training set and thus improving its capacity for generalization. In an adaptive array, this effect is analogous to allowing the adaptive processor to forget previously detected signals after an amount of time about three times the decay time constant. Such an effect is useful when the input signals are not wide sense stationary (WSS). For example, when beam-forming towards a moving target, weight decay ensures that the antenna directivity pattern will not point towards previous positions of

Learning curve.

MSE (a.u.)

Figure 4.38: Mean squared error during binary chirp adaptation.

Array Element

Taps

Figure 4.40: Final weights after adaptation, showing a thin and straight correlation slice.

Angle of Arrival (radians)

Frequency (kHz)

Figure 4.39: Array receptivity pattern in the presence of a binary input signal.

Spatial frequency (1/m)

Temporal Frequency (kHz)

Figure 4.41: Fourier transform of the final weights with non-linear terms symmetrically positioned around the desired signal.

created regions of high receptivity at higher frequencies (> 2.5 kHz) and at high AOAs (> 0.8 radians). The origin of these regions is better understood if we look at the final weights after adaptation (Fig. 4.40). The weights adapt to the auto-correlation of the desired binary chirp. However, the processor detects the aliased non-linear terms folded

where $\mathbf{w}^o$ is the optimum Wiener solution (Eqn. 1.7) and $\mathbf{w}^d$ is the solution with weight decay. Diagonalizing the auto-correlation matrix, we have

$$\mathbf{v}^d = \left(\mathbf{I} + \Lambda^{-1}\frac{(1-Y)}{\eta'}\right)^{-1}\mathbf{v}^o,$$ (4.12)

where $\Lambda = \mathbf{Q}^t\mathbf{R}\mathbf{Q} = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_{NT})$, and $\mathbf{v}^o = \mathbf{Q}^t\mathbf{w}^o$. This uncouples the equations, and equation 4.12 can be solved for the individual elements v_i of $\mathbf{v}^d$, providing the solution

$$v_i^d = v_i^o\left(\frac{1}{1 + \frac{(1-Y)}{\eta \lambda_i}}\right),$$ (4.13)

where v_i^o is an element of $\mathbf{v}^o$. Some interesting conclusions can now be drawn. First, the larger the learning rate, the closer the solution v_i will be to the optimum v_i^o. This should be expected, since weight decay modifies the LMS update equation (Eqn. 2.27) by

$$w_{ji,\tau}(\kappa+1) = Y w_{ji,\tau}(\kappa) + \eta' e_j(\kappa) s_i(\kappa - \tau).$$ (4.14)

Thus, a high learning rate allows the weights to build up more quickly, compensating for their continuous decay. However, the dynamic weight equation becomes

$$\mathbf{w}(\kappa) = \sum_{l=1}^{NT} \mathbf{C}_l(Y - \eta'\lambda_l)^\kappa + \mathbf{R}^{-1}\mathbf{p},$$ (4.15)

and for stability (all poles inside the unit circle), the learning rate must be bounded so that

$$-1 < Y - \eta'\lambda_i < 1,$$ (4.16)

for every λ_i, resulting in the following boundary condition:

$$\frac{Y-1}{\lambda_{min}} < \eta' < \frac{Y+1}{\lambda_{max}},$$ (4.17)

where λ_{min} and λ_{max} are the minimum and maximum eigenvalue of $\mathbf{R}$, respectively. Thus, weight decay can not be compensated for by using an arbitrarily large learning rate. Note that when there is no decay, $Y = 1$, and equation 4.17 reduces to the typical boundary condition ($0 < \eta < 1/\lambda_{max}$) [96] (remember that $\eta' = 2\eta$). The second

---

Figure 4.42: (Top) The desired signal used in all of the simulations in this section: a binary broadband down-chirp. (Bottom) The spectrum of the desired signal.

the moving target. In order to simulate a simplified version of the PR decay, a feature was added to the simulation which allowed every weight to be multiplied by a constant value Y of less than one at every iteration. This constant can be calculated from the PR time constant τ_{pr} by using the relation

$$Y = \exp\left(-\frac{\Delta_t}{\tau_{pr}}\right),$$ (4.9)

where Δ_t is the temporal sampling rate (the period of time between SLM frames). When weight decay is taken in consideration, the weight adaptation equation (Eqn. 4.4) becomes

$$\Delta\mathbf{w} = -\eta'(\mathbf{R}\mathbf{w} - \mathbf{p}) + (Y-1)\mathbf{w}.$$ (4.10)

Solving for $\mathbf{w}$, we have

$$\mathbf{w}^d = \left(\mathbf{I} + \mathbf{R}^{-1}\frac{(1-Y)}{\eta'}\right)^{-1}\mathbf{w}^o,$$ (4.11)

require a larger dynamic range of the weights in order to represent their correlation.

The minimum weight amplitude is limited by noise — set at 15 dB below the power of the desired signal in these simulations — while the maximum amplitude is limited by weight decay, since the weights with highest amplitude suffer the most decay. Thus, weight decay has the effect of reducing the dynamic range of the weights and, therefore, limits their capacity to reproduce the best transfer function for signal detection and jammer cancellation.

Since the dynamic range of the weights is smaller, one should also expect that a smaller number of iterations would be necessary to reach the saturation point — where decay just offsets learning. Learning should require a smaller number of iterations until convergence, not because learning becomes faster, but because learning terminates sooner. Figure 4.44 shows the learning curve obtained when beam-forming in the direction of the same broadband chirp described in section 4.3.1, with the same learning rate. Note that only 2000 iterations are necessary for $Y = 0.999$. The maximum magnitude of the weights after adaptation is 1.6 times smaller than that achieved with no decay in the simulation, but the final MSE is almost three orders of magnitude worse.

Figure 4.45 shows the output signal in time (top) and frequency (bottom). Note that the detected bandwidth is slightly reduced compared to the original signal bandwidth. This effect can also be attributed to the limitation on the maximum dynamic range of the weights due to weight decay, since a broadband FIR filter must have a sharp peak in its correlation weights.

4.3.3 Near-field signals and CW jammer nulling

The diagram in figure 4.46 depicts the detection of a binary chirp signal in the near-field. Notice that the source emits spherical wavefronts that, in the near-field, result in a quadratic relative time-delay between antenna elements. The relative time-delay between antenna elements and the resulting shape of the correlation slices is discussed in

Figure 4.43: Weights after adaptation, showing a thin correlation slice with small side-lobes.

Figure 4.44: MSE with weight decay, showing faster convergence.

conclusion is that, the larger the PR time constant, the closer Y will be to one, and the closer the final weights will be to the optimum solution. Note that if Y and η' are within bounds ($0 < Y < 1$ and $\frac{Y-1}{\lambda_{min}} < \eta' < \frac{Y+1}{\lambda_{max}}$), equation 4.14 always has a solution (assuming $\mathbf{R}$ to be positive definite, which is a reasonable assumption in the presence of some white noise in the input [90]).

The third conclusion is the most interesting one: the larger the eigenvalue λ_i, the less affected v_i is by weight decay. This means that strong CW jammers in the far-field are still nulled efficiently. However, it becomes more difficult to detect and null broadband signals, specially in the near-field, since the energy spectrum of the signal will be spread out over a number of eigenvalues. This conclusion can also be reached by a different argument. From equation 1.23, the weights converge to the cross-correlation between the input signal and the error signal. The larger the bandwidth of the signals involved, the sharper and narrower the peaks of their correlations, and the smaller their side-lobes (Fig. 4.43). Narrowband signals, on the other hand, have broader auto-correlations with more of its power distributed to its side-lobes. Thus, broadband signals

Figure 4.46: Detection of a broadband chirp in the near-field. The curved wavefronts result in a quadratic time-delay between antenna elements

of the input.

The simulation ran for 7680 iterations, requiring about 10 minutes of computational time, or 960 ms of simulated time, after which the MSE converged to its minimum value of 0.232, as shown in figure 4.49. After adaptation, the receptivity pattern shown in figure 4.50 is obtained, showing a strong response at 0.3 radians (the AOA of the desired chirp), with a spread of 0.2 radians about its center. A strong null is present at 1.5 kHz, spreading in angle throughout the signal spectrum even though the main beam is well resolved. That is because the side-lobes of the CW jammer dominate the signal within this frequency beam.

The weight matrix converges to the values graphed on figure 4.51, where the correlation band of the broadband chirp is clearly visible. Note that this band is curved, as should be expected, since for a near-field signal, the propagation delay from the signal source to the antenna array varies quadratically from antenna element to antenna element. Figure 4.52 shows the weights in the Fourier domain. The narrow and curved correlation slice occupies a region in the Fourier spectrum equivalent to that occupied by the input signal, indicating that the antenna array has a strong receptivity for signals occupying that region in the spectrum. At the frequency of the jammer, however, there is a long vertical null, reflecting the system's weak response to that frequency at all directions. Note, however, that there is some response at the position in spectrum

Figure 4.45: (Top) The output of the system with weight decay. (Bottom) The spectrum of the output.

Figure 4.45: (Top) The output of the system with weight decay. (Bottom) The spectrum of the output.

more detail in section 5.5. In this example, the signal of interest is a broadband binary chirp identical to the one described in section 4.3.1, but this time its source is located in the near-field of the antenna array, at an angle of 0.3 radians and at a distance of 4.18L from the center of the array, where $L = N a$ is the total length of the array. Also, a narrowband jammer (not shown in Fig. 4.46 for clarity) with twice the power of the desired chirp is present in the far-field, with an AOA of -0.5 radians, as depicted in figure 4.47. Figure 4.48 shows the same signals in the spatio-temporal Fourier plane. Note that the CW jammer is clearly distinguished as a strong point at 1.5 kHz. The near-field chirp, however, has its power spread over a pie-shaped region of the Fourier spectrum. This is because a broadband signal in the near-field occupies a spread of angles-of-arrival, each of which contributing to a portion of the spectrum shown in the figure. Also note other weaker pie-shaped regions. Those are the result of the aliased spectral fold-over of the higher order non-linear terms due to the binary quantization

Figure 4.52: Weight matrix in the spatio-temporal Fourier domain for a near-field signal and CW jammer.

Figure 4.51: Weight matrix after adaptation for a near-field signal and CW jammer.
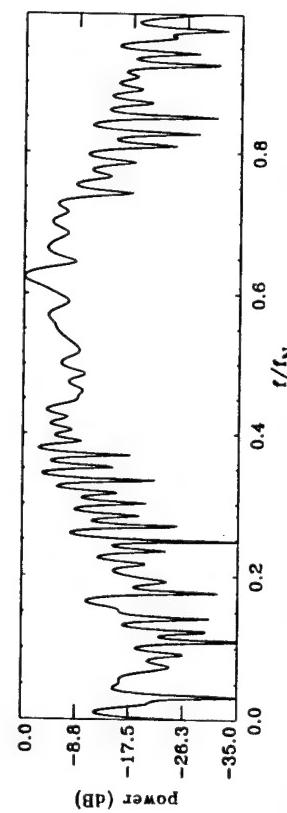
Figure 4.53: (Top) The output of the system in the presence of a CW jammer. (Bottom) The spectrum of the output.

Fourier spectrum of the output is shown in the bottom graph, showing that the whole signal spectrum has been recovered.

Figure 4.48: Near-field chirp and CW jammer in the Fourier domain.

Figure 4.47: Input signals: a near-field broadband chirp and CW jammer.

Figure 4.50: Receptivity pattern in the presence of a near-field chirp and a narrowband jammer.

Figure 4.49: Learning curve in the presence of a near-field chirp and a narrowband jammer.

corresponding to the jammer's frequency and AOA. This shows us that the system uses some of the jammer's power to null itself.

The system output is depicted in the top graphic in figure 4.53. Note that the desired broadband chirp is completely recovered, asides from some of the noisy behavior at the highest and lowest frequencies attributed to the effects of weight decay. The

Figure 4.56: Fourier spectrum of a broadband chirp, two broadband jammers and a CW jammer detected by a circular array.

Figure 4.55: Input signals for the circular array: a broadband chirp, two broadband jammers and a narrowband jammer.

the signal.

The system adapted for 12800 iterations, which corresponds to about 15 minutes of real time, or 1.6 s of simulated time. Figure 4.57 depicts the final weights after adaptation, showing two thin sinusoidal correlation slices with peaks at the antenna elements corresponding to the AOAs of 0.25 radians and -2.0 radians, indicating that the system has a strong response for both of the desired signals coming from those angles of arrival. In the Fourier domain the weights display a spectrum similar to the spectrum of the input signals, as shown in figure 4.58.

Figure 4.59 depicts the output of the system after adaptation, showing that the broadband binary chirp has been recovered. Note that even though two chirps are present at the input, they are coherently combined in order to provide a single chirp at the output. The bottom graph shows that the chirp has been recovered over its entire spectrum.

Figure 4.54: Simultaneous detection of two broadband chirps in a circular array. The linear rastering of the antenna elements results in a sinusoidal relative delay between antenna elements.

4.3.4 Multiple beam-forming in a circular array

This scenario illustrates two points of interest: the use of circular arrays and the possibility of beam-forming towards multiple desired signals. In a circular array, the antenna elements are evenly spaced in a circle separated by an angle $\gamma = 2\pi/N$. Figure 4.54 depicts this scenario, where the signals from the antenna elements are linearly rastered before being fed to the scrolling SLM, resulting in a sinusoidal relative time-delay between antenna elements. The relative time-delay between antenna elements and the resulting shape of the correlation pattern are discussed in more detail in section 5.5. Figure 4.55 shows the input signal as seen by the antenna array when detecting two broadband chirps of equal amplitude (the same matrix in the center of Fig. 4.54), the first with an AOA of 0.25 radians and the second with an AOA of -2.0 radians, depicting two overlapping sinusoids with a shift of 2.25 radians between their peaks. Figure 4.56 shows the corresponding spatio-temporal Fourier spectrum of the detected signals. Since each antenna element sees the broadband input arriving at a different AOA, the corresponding spectrum occupies all possible AOAs over the bandwidth of

Figure 4.61: Fourier spectrum of a broadband chirp and a broadband jammer detected by a circular array.

Figure 4.60: Input signals: a broadband chirp and a broadband jammer with twice the chirp's power, detected by a circular array.

better understood through simulations. The input signals consist of a binary chirp in the far-field with an AOA of 0.2 radians and a broadband jammer with twice as much power as the chirp and with frequencies ranging from 1 kHz to 3 kHz (the same range as that of the chirp), at an AOA of -0.35 radians (Fig. 4.60). Figure 4.61 shows the corresponding Fourier spectrum. Note that, once again, the signals occupy all AOAs, as expected from far-field signals detected by a circular array.

Adaptation takes place over 12800 iterations, providing the learning curve shown in figure 4.62, describing a well behaved adaptation. However, the array receptivity pattern (Fig. 4.63) shows otherwise. Note the presence of two AOAs of strong array response: one at 0.2 radians (the AOA of the desired signal) and another at -0.35 radians (the AOA of the jammer), where a null should be expected. This is a consequence of the use of a broadband jammer which is repeated over a limited time window. There are only 256 time samples available for a processor with 128 time-delays. Thus, the signals (desired and jammer) have to be repeated after every 256 time-steps, and the

Figure 4.58: Weight matrix in the Fourier domain, showing that AOA info is not readily available.

Figure 4.57: Weight matrix displaying two sinusoidal correlation slices.
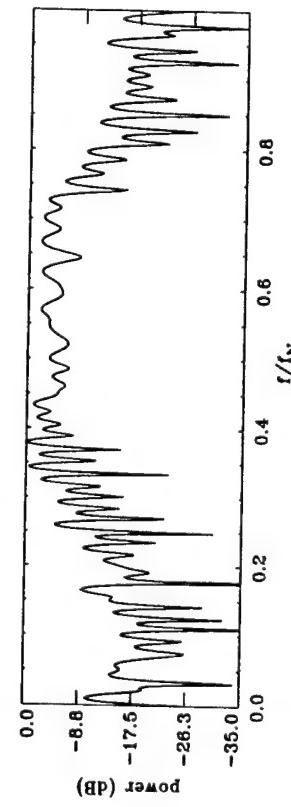
Figure 4.59: (Top) The output of the system after beam-forming towards two desired beams in a circular array. (Bottom) The spectrum of the output.

4.3.5   Broadband jammer nulling

This scenario illustrates broadband jammer nulling in a circular array. It describes a limitation in signal representation of our experimental implementation which can be

Figure 4.65: Receptivity pattern of a circular array in the presence of an uncorrelated broadband jammer. The jammer gets nulled.

Figure 4.64: Learning curve in the presence of a broadband jammer in a circular array using an input matrix with 2048 time samples.

Figure 4.63: Receptivity pattern of a circular array in the presence of a repeating broadband jammer. The jammer is linearly transformed into the desired signal instead of being nulled.

Figure 4.62: Learning curve in the presence of a repeating broadband jammer in a circular array.

Unfortunately, the experimental processor presented in chapter 5 is limited to 128 time samples due to memory and communication limitations of the SLM driver. Note, however, that this corresponds to an experimental limitation to the simulation of the arrival of continuous data. It is not a limitation of the adaptive capacity of the processor. Such an effect could be fixed with the use of a PCI controlled SLM driver, which has been long promised by its manufacturer, but has not been released until the date of publication of this thesis. Moreover, as will be seen in chapter 5, the simulations are capable of accurately describing the behavior of the experimental setup, showing that the relevant physical details are incorporated in these simulations. Figure 4.66 shows the output obtained with a repeating jammer for comparison with the experimental results obtained in chapter 5. Since the broadband jammer has been linearly transformed to the desired chirp, it is no surprise that the binary chirp is recovered at the output over its entire bandwidth. Note, however, that such an effect can not take place when the jammer is narrowband, since no linear transformation is capable of producing signals

jammer ends up being partially correlated with the desired signal. Consequently, the processor is capable of adapting the weights in order to produce a linear transformation that correlates the jammer into the desired output signal. If instead the jammer had been continuously generated and not repeated, it would be truly uncorrelated with the desired signal and the processor would have no choice but to cancel it out. To illustrate this point a second simulation was generated with the same input signals, but this time using 2048 samples to generate the input instead of 256. That is, the input signals are not repeated until after 2048 samples, decreasing the amount of correlation between jammer and the desired signal. The resulting learning curve is shown in figure 4.64. Note that the minimum MSE (0.33) is not as low as it previously was (0.20), since the processor is no longer capable of linearly transforming the jammer into the desired signal. However, the receptivity pattern (Fig. 4.65) behaves as expected, displaying a strong and narrow response at the AOA of the desired signal, while nulling a broadband signal at the AOA of the jammer.

Figure 4.68: Fourier spectrum of a broadband chirp, two broadband jammers and a CW jammer detected by a linear array.

Figure 4.67: Input signals: a broadband chirp, two broadband jammers and a narrowband jammer.

Figure 4.70: Receptivity pattern showing the nulling of the CW jammer and adaptation to broadband jammers.
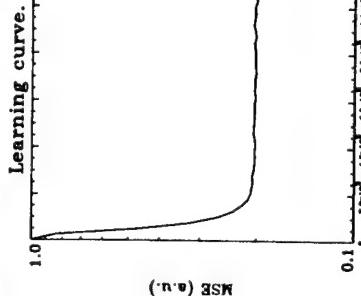
Figure 4.69: Learning curve in the presence of two repetitive broadband jammers and a CW jammer.

figure 4.69. The system adapts in the presence of repetitive broadband jammers, which are partially correlated with the desired signal and, consequently, produce a non-zero response at the AOA of the broadband jammers, as depicted in figure 4.70. Note that the CW jammer, however, has been nulled over all spatial frequencies.

Figure 4.66: (Top) The output of the system in the presence of a repetitive broadband jammer but no desired signal. (Bottom) The spectrum of the output.

outside the frequency of the CW jammer.

4.3.6 Far-field beam-forming and multiple jammer nulling

This final simulation illustrates the processor behavior in the nulling of multiple jammers. A linear array is chosen this time, with all signals placed in the far-field.

The desired signal consists of the same broadband chirp used before, with an AOA of 0.25 radians. A narrow-band jammer with as much power as the signal is placed at an AOA of -0.5 radians. Two broadband jammers are used, each with as much power as the signal (for a total jammer power three times larger than the signal power), the first one occupying the same frequency range as the desired chirp, at an AOA of 1.0 radian, and the second one with frequencies ranging from 1.5 kHz to 4 kHz, at an AOA of -1.2 radians (Figs. 4.67 and 4.68).

The system adapts for 12800 iterations, following the learning curve shown in

Figure 4.71: Weight matrix displaying a thin and tilted correlation band.
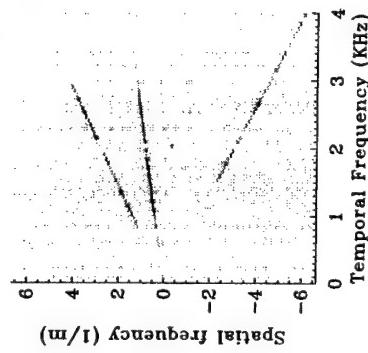
Figure 4.72: Weight matrix in the Fourier domain.

Figure 4.71 shows the weights after adaptation, displaying the expected auto-correlation curve with a tilt corresponding to the time-delay necessary to coherently sum the far-field desired chirp. Its Fourier spectrum is shown in figure 4.72, showing a null at the frequency of the CW jammer and the detection of the other two broadband jammers.

The system output is depicted in the top graph of figure 4.73, showing that the binary chirp has been recovered. The lower graph, however, shows that some of the CW jammer is still present at 1.5 kHz.

4.4 Summary

This chapter presented simulations and analysis concerning the application of the linear FIRNN to the adaptive array problem. While the first set of simulations dealt with general adaptive array problems using the BEAMTAP algorithm, the second set addressed issues concerning the actual optoelectronic implementation of a binary input time-delay-and-sum LMS array with weight decay.

Some important conclusions have been reached. Those are:
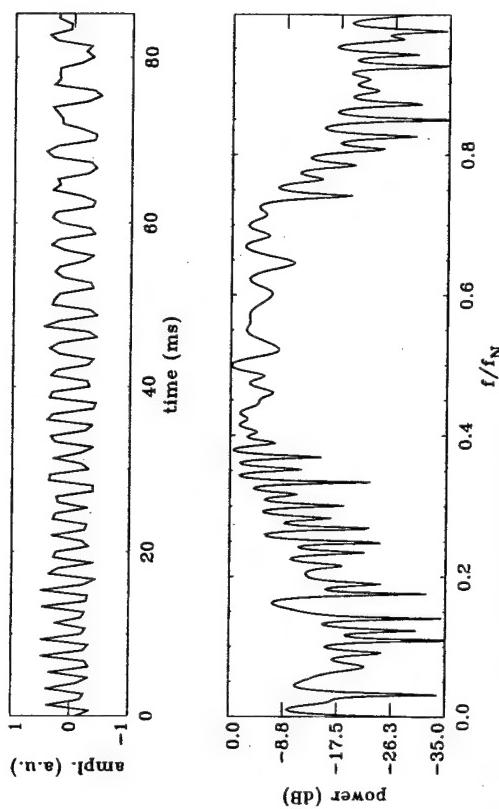
Figure 4.73: (Top) The output of the system in the presence of two broadband jammers and one CW jammer. (Bottom) The spectrum of the output.

(1) The dynamics of weight adaptation are driven by the eigenvalues of the auto-correlation matrix $\mathbf{R} = E[\mathbf{ss}^T]$.

(2) The performance of an adaptive array can always improve with the number of taps in its delay-lines, and one can devise examples in which the optimum time-delay is much larger than the time-of-flight of the incoming input wave over the antenna array.

(3) The LMS adaptive array is capable of nulling multiple broadband jammers and its performance is reduced gracefully with the number of jammers, except when in-band main-beam jammers are present, in which case the performance deteriorates rapidly. The larger the resolution of the array the less likely it is to have jammers falling over the same temporal and spatial frequency bins of the desired signal.

(4) A linear adaptive array is capable of adapting for the detrimental effects of non-linearities. However, the available number of degrees of freedom is reduced, and one can devise instances in which the performance of the array degrades rapidly (such as main beam jamming).

(5) Binary quantization of the input signal can give rise to input non-linearities, but it does not prevent the processor from successfully adapting.

(6) Weight decay causes a reduction in the dynamic range of the weights, limiting the array performance. The larger eigenvalues of the auto-correlation matrix, however, are less affected by the decay.

Finally, some examples of CW and broadband jammer nulling in linear and circular arrays have been presented. Those examples are useful when compared to the actual experimental results, as presented in chapter 5.

Chapter 5

Experimental demonstration of an input delay plane architecture single-layer FIRNN

5.1 Introduction

As explained in chapter 2, the single-layer, linear FIRNN is functionally equivalent to a true-time-delay adaptive array that utilizes the LMS algorithm for weight adaptation [18]. The goal of this chapter is to demonstrate an experimental implementation of an optoelectronic processor using the input delay plane architecture described in chapter 3 applied to the adaptive array problem at sonar frequencies. Since the adaptive array processor presented here is derived from a study which originated from optical FIRNNs, it has been termed as the Finite Impulse Response Optical Neural Network (FIRONN) [97].

Real-time adaptive processing of wide-band signals in large antenna arrays is an extremely demanding signal processing task, since wide-band signals require the use of a large number of time-delays in order to avoid beam-squint (the variation of the AOA of a broadband signal with frequency when detected by a phased antenna array without true-time-delay). This task becomes specially difficult when the signals are present in the near-field of the array, or when the array shape is not planar or is time varying, as these scenarios require a higher number of degrees of freedom and, therefore, more processing capacity. Several photonic implementation schemes have been previously proposed [68, 70, 58] and discussed in chapter 1. However, we believe that this is the

first experimental implementation of an adaptive array processor capable of full spatio and temporal beam-forming and simultaneous multiple jammer nulling of broadband signals from up to 256 antenna elements.

In the following implementation, a scrolling SLM is used for representing the input signal and its delayed versions, operating as a multi-channel delay-line. Dynamic holographic grating formation in a photorefractive crystal is used for representing the adaptive weights. A temporal multiplexing scheme is adopted for read/write isolation, and differential heterodyne detection is used for bipolar signal detection. For persistent interferometric stability of the heterodyne detector, an active phase locked-loop is used to control a piezo-actuated mirror in the path of the heterodyne reference beam. Section 5.2 describes the architecture being used and its optoelectronic implementation. Section 5.3 presents experimental results for five distinct scenarios, representative of the capabilities of the processor. Many of these scenarios are identical to the ones used in the simulations presented in section 4.3 of chapter 4, allowing the comparison between experimental and simulation results, verifying the expected performance of the optoelectronic processor.

5.2 Optical architecture

Figure 5.1 shows the schematic diagram of the optoelectronic architecture used in the processor. It makes use of temporal integration for weight adaptation and spatial integration for output computation. The input signal array $g(\kappa)$ is used to modulate the first row of the scrolling SLM. At every time step, the input vector is shifted down by one row of the SLM, thereby producing a spatio-temporal plane representing the delayed versions of the input vector. Thus, input element s_i delayed by τ time steps will be represented by the SLM pixel located at position (i, τ) of the spatio-temporal input plane. The complex amplitude distribution represented by this spatio-temporal plane is diffracted by the dynamic holographic gratings stored in a PR crystal, where
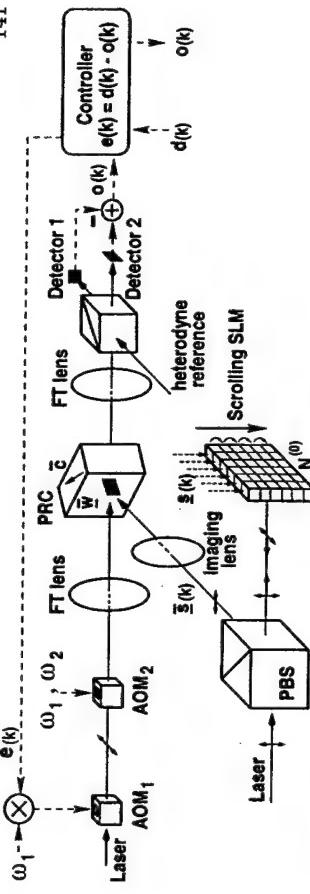
Figure 5.1: Optoelectronic architecture for a sonar adaptive array processor using a scrolling SLM for delayed input representation. Dynamic gratings recorded in a volume hologram are used for implementing the adaptive interconnections, followed by spatial integration onto the output detectors. Differential heterodyne detection is used for bipolar signal representation. The output signal is electronically amplified and subtracted from a desired signal, producing a feedback error term which modulates a laser beam using an AO modulator. A second modulator compensates for the Doppler shift of signal frequencies in the first modulator and brings the signal back to baseband, while simultaneously introducing a second tone, ω_2, used for phase-locked loop stabilization.

each grating represents one of the adaptive weights. The diffracted output is spatially integrated by a lens onto the output detectors, thus performing the correlation operation $o(\kappa) = \sum_i \sum_\tau w_{i,\tau}\, s_i(\kappa - \tau)$. Heterodyne detection is used in order to achieve high detection gain and, more importantly, a bipolar representation for the output signal. Differential detection is used in order to better cancel the DC term of heterodyne detection, while reducing common mode laser amplitude noise. During processing, the output signal is digitized and subtracted from the desired signal using a micro-controller, which also controls the temporal read/write multiplexing. The error signal is mixed with a carrier frequency ω_1, at the central frequency of operation of the first acousto-optic modulator, which modulates an incident laser beam. The second AOM is modulated with a sum of tones, ω_1 and ω_2, which are slightly offset with respect to each other. The first tone (ω_1) compensates for the Doppler shift imposed by modulation on the first AOM, bringing the error signal modulation back to baseband, while the second
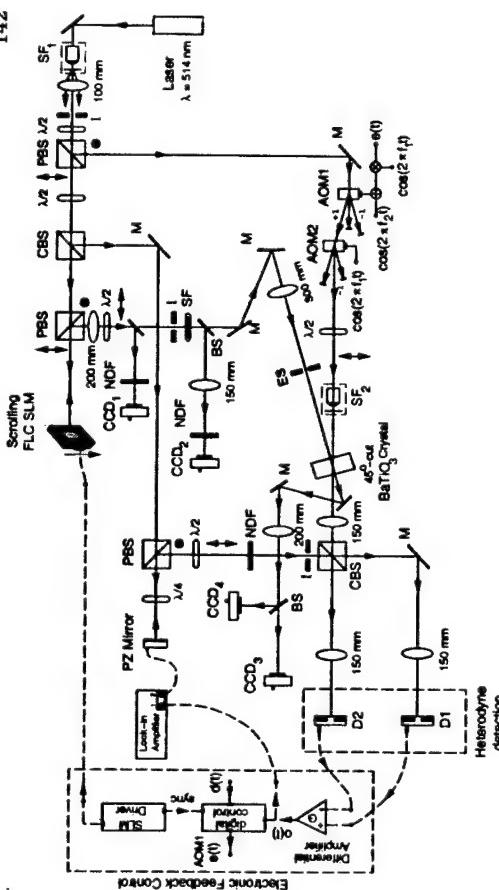
Figure 5.2: Detailed experimental diagram: M – mirror, BS – beam-splitter, CBS – cubic beam-splitter, PBS – polarizing beam-splitter, SF – spatial filter, NDF – neutral density filter, D – detector, PZ – piezo-electric, CCD – charge coupled device, SLM – spatial light modulator, FLC – ferro-electric liquid crystal, AOM – acousto-optic modulator. Solid lines represent optic signals and dashed lines represent electronic signals.

Figure 5.3: Photograph of the optical setup showing the FLC-SLM used as a scrolling SLM, the second AOM used to modulate the error feedback signal, the PR crystal used to record the dynamic holographic gratings during adaptation, the piezo-mounted mirror used in long-term heterodyne stabilization, and the two detectors used in bipolar heterodyne detection. The beam path is described by a gray arrow.

tone (ω_2) produces an intermediate frequency (IF) tone at $\omega_1 - \omega_2$, which is used to measure the phase drift between the two arms of the heterodyne detector using a lock-in amplifier. The optical beam modulated with the error signal is then used to modify the index grating formed in the PRC, by interfering it with the spatio-temporal input plane $\overline{s}(\kappa)$, represented by the scrolling SLM.

Figure 5.2 shows an experimental schematic diagram of the processor being implemented. A ferro-electric liquid crystal (FLC) SLM is used as a scrolling SLM. A 45°-cut BaTiO$_3$ photorefractive crystal is used for recording the dynamic hologram. A lock-in amplifier is used to detect the angle of deviation between the two arms of the heterodyne detector and to generate a signal proportional to this deviation, moving a mirror mounted on a piezo actuator in order to compensate for variations in the optical path. Four CCDs are used to monitor the processor performance: CCD$_1$ displays the

amplitude distribution of the scrolling SLM in the Fourier plane while CCD$_2$ displays it in the image plane. CCD$_3$ records the image diffracted by the dynamic holographic grating recorded in the PR crystal, corresponding to an image proportional to the magnitude of the adaptive weights, while CCD$_4$ displays the same image in the Fourier plane. In the following sections the operation of the devices and sub-systems used in the optoelectronic processor is discussed in detail .

5.3 Sub-system description and analysis

This section presents a physical and analytical description of each of the sub-systems used in the FIRONN optoelectronic processor.

5.3.1 The scrolling SLM

A Displaytech FLC-SLM with 256×256 pixel elements is used as a scrolling SLM. It is controlled by an electronic driver which can store up to 128 frames and is capable of displaying them in succession at frequencies from 2 Hz up to 2.5 kHz. The scrolling of the input signal was simulated by first generating the data frames in a computer, then generating the 128 frames representing their sequential animation and transferring all the frames to the SLM driver. Thus, a "movie" is generated, which simulates the arrival of a new input data vector at each time step, representing the instantaneous signal detected by the antenna array. The frames are generated such that the 129th frame is identical to the first, thereby enabling the continuous simulation of the arrival of data without interruption into $N = 256$ delay lines, each with $T = 127$ time delays. This approach, however, is only capable of representing periodic signals with periods of 128 or its sub-multiples (64, 32, 16, etc.) whereas a true scrolling SLM would be capable of representing real-time signals from the array.

The SLM is used as a binary phase modulator: horizontally polarized light is normally incident on the SLM and is reflected back to a polarizing beam-splitter (PBS). The liquid crystal used to fill the SLM pixels have their fast axis oriented at 22.5° (-22.5°) with respect to the incident polarization in their off (on) state, such that, after propagation through the liquid crystal, reflection at the back reflecting surface, transmission through the same medium, and reflection off the PBS selecting the vertical component results in a 180° phase difference between the on and off states. This modulation configuration reduces the amount of power left unmodulated, if compared to

amplitude modulation. However, since the fill factor of our SLM is limited to 87% and the PBS is far from being an ideal polarizer, there is inevitably some power left at DC. Thus, spatial filtering is used in the Fourier plane of the SLM in order to remove the central DC spot, since the optical power left at this spot conveys no information and it would only cause a reduction in the output signal-to-noise ratio (SNR) by contributing to the decay of the PR grating and increasing the amount of optical power in PR beam fanning [63]. The negative spatial frequencies are also removed, since it is easier to perform spatial filtering by removing both DC and negative side-bands with a razor blade rather than removing only DC. For a scrolling SLM, these negative spatial frequencies also are at a negative temporal frequency, resulting in single side-band temporal modulation. The final intensity of the SLM beam reaching the PRC is measured to be about 300 mW/cm².

FLC-SLMs operate in a bistable mode, in which an applied field overcomes the elastic and surface energy barrier of the liquid crystal smectic layers, thereby rotating the molecules in the smectic plane, changing the optic axis of the crystal by 2ϕ, where ϕ is the smectic tilt angle at the surface [82]. As a consequence of bistability and the bidirectional drive, fast switching is achieved (as fast as 10 μs), but the device is limited to binary operation. This, however, is compensated by the fact that 256×256 individually addressable pixels are available, providing us with enough processing gain that good performance is achieved even when the ratio between the input signals is such that the power of the weakest signal falls below the power of the quantization noise.

Another issue is the dead time between frames. It takes the driver a fixed amount of time ($t' = 0.36$ ms) to load a new frame. The partial image displayed during this time is considered invalid and, therefore, should not contribute to the holographic recording. This problem is solved by using the micro-controller to gate the error signal going in the feedback AOMs, such that the error signal is always off when the SLM is displaying invalid data.

The optical field amplitude of pixel (i,τ) after SLM modulation can be represented by

$$E^A_{i,\tau}(\kappa;x,y,z,t) = C_s\, e^{j(\omega t - kz)}\,\text{rect}\left(\frac{x - i\Delta_x}{\Delta'_x}\right)\text{rect}\left(\frac{y - \tau\Delta_y}{\Delta'_y}\right)$$
$$\times \text{rect}\left(\frac{t - (\kappa-\tau)\Delta_t}{\Delta'_t}\right) s_i(\kappa-\tau), \qquad (5.1)$$

where $C_s = \alpha_1 \sqrt{\eta_s}\, E_0$, where α_1 is the fraction of the initial field amplitude E_0 incident on the SLM after beam-splitting, η_s is the modulation efficiency of the SLM, $\text{rect}(\cdot)$ is the unit rectangular function, Δ_x and Δ_y are the pixel pitches in the x and y directions, while Δ'_x and Δ'_y are the respective pixel dimensions. In our case, $\Delta_x = \Delta_y = 15\ \mu m, \Delta'_x = \Delta'_y = 14\ \mu m$. Δ_t is the time interval between time steps, determined by the frame rate selected on the SLM driver. We have set it at 1.5 kHz and, therefore, $\Delta_t = 0.667$ ms. Thus, the current time step is given by $\kappa = t$ DIV Δ_t, where DIV denotes integer division. $\Delta'_t = \Delta_t - t' = 0.31$ ms is the time during which the SLM contains a valid image. $s_i(\kappa-\tau)$ represents the zero-mean signal detected by the antenna element i after sampling and quantization, assuming the values -1 and 1. In discrete notation, the field amplitude from pixel (i,τ) is given by

$$E^A_{i,\tau}(\kappa) = C_s\, s_i(\kappa-\tau). \qquad (5.2)$$

The total modulated field is given by

$$E^A(x,y,z,t) = \sum_{i=1}^{N}\sum_{\tau=0}^{T} E^A_{i,\tau}(x,y,z,t). \qquad (5.3)$$

5.3.2 Temporal multiplexing and digital control

Referring to figure 5.2, the input plane is diffracted by the adaptive PR hologram to the output detectors. The diffracted output produced by the hologram travels along exactly the same optical path as the error beam in the AOM arm. Thus it is necessary to distinguish the diffracted output from the error beam from the AOM arm, used to record the hologram. Different techniques for read-write multiplexing include

wavelength, polarization [84] Bragg degenerate angle [85], hybrid angle-polarization in a uniaxial crystal [86], coherence multiplexing, frequency multiplexing and time multiplexing [41]. Since we are already using a micro-controller for calculating the error signal, we decided to also use it to control the temporal multiplexing of the read and write operations, using the SLM driver trigger output for synchronizing the operation. At first, time multiplexing was attempted at the frame rate of the SLM driver (Δ_t). However, this approach presented the following problem: the detector and differential amplifier require some time to recover from saturation in the write cycle, until a clean signal is present in the read cycle. There is a limited amount of time Δ'_t available for all operations (write, recover and read), and the frame rate would have to be limited to less than 1 kHz in order to accommodate them all in one time period Δ'_t. Thus, it was decided to perform time multiplexing using equal blocks of time for each operation, denoted by read cycle and write cycle. During the read cycle [$t_R = (T+1)\Delta_t = 85.4$ ms] no signal is sent to the AOM arm, thus shutting off that beam. Consequently, the diffracted signal can be detected by the heterodyne detector without interference of the straight-through beam from the AOM arm. Meanwhile, the error signal is calculated and stored in the micro-controller memory at each time step. Next, the write cycle ($t_W = t_R = 85.4$ ms) is initiated, during which the output of the detectors is blocked out from the high-gain amplifiers by a fast electronic switch, and the stored and delayed error signal is fed to the AOM as the SLM repeats its scrolling loop so that the interference between the SLM and AOM beams in the PRC modifies the recorded index grating in the PRC.

This scheme results in a delay of $T+1$ time steps between the calculation of the error signal and its use in feedback. Such a delay is allowable because of the periodicity of the signal which we are feeding through the scrolling SLM [$s_i(\kappa - T - 1) = s_i(\kappa)$]. Since the PR time constant ($\tau_{pr} \approx 0.4$ s) is larger than the read and write cycles, the grating exponential erasure is kept small during these periods of time, and algorithmic

Figure 5.5: Picture of the digital control sub-system, showing the opposite side of the HC11 evaluation board where the BNC connectors, DAC chip, op-amp and unit gain buffers have been soldered.

sate for any fluctuations in the 2.5 V offset during the A/D conversion. The digitized output and the calculated error signal are made available at a data bus through output data ports B and C, respectively. The data bus consists of a 40 pin parallel ribbon cable connector in which all the lower pins are signal grounds and the 20 upper pins are wired to ports B, C and the four most significant bits (MSB) of port A, as indicated in figure 5.4. It connects the digital controller to a personal computer (PC) via the AT-MIO-16DE10 data acquisition (DAQ) board [100] programmed by the LabView software package [101]. Note, however, that the DAQ board and the PC are present in order to program the micro-controller and to provide a second way for collecting output data (besides probing signal $O(k)$ with an oscilloscope), and their presence is not required for the normal operation of the adaptive processor.

The D/A (digital-to-analog) converter (DAC0800) and the LM741 op-amp con-

Figure 5.4: Schematic diagram of the digital control section of the experimental setup. The SLM driver provides the SYNC signal used to synchronize the HC11 micro-controller. The output signal $O(k)$ is digitized at every time step during the read cycle. It is subtracted from the desired signal stored in its internal EEPROM and the resulting error signal $E(k)$ is stored in RAM until the next write cycle. The D/A converter (DAC0800) and the LM741 op-amp convert the digital output into an analog waveform. The LM6221 unit gain buffers drive the 50 Ω lines used to drive the AO modulators [$E(k)$] and a signal used to indicate the sate of the current cycle ($R\overline{W}$).

convergence is not compromised (for the effect of decay on the weight convergence, see Section 4.3.2).

The diagram shown in figure 5.4 depicts the digital control section of the experimental setup. The SLM driver provides the SYNC signal, which is connected to one of the edge detecting strobe inputs (pin 0) of port A in the micro-controller, a Motorola M68HC11E9 [98, 99], (HC11). During the 128 time steps of the read cycle, whenever a rising edge is detected, the output signal $O(k)$ is read out through the A/D (analog-to-digital) input port E (pin 1) and it is subtracted from the desired signal, which has been previously programmed in the internal EEPROM of the micro-controller. The resulting 128 error signals $E(k)$ are stored in RAM until the next write cycle. Its mean value is calculated and subtracted from each sample in the time-series, in order to compen-

Figure 5.6: Synchronized readout and A/D conversion. The analog output signal (solid line) is read and digitized soon after the rising edge of the SYNC signal (dashed line). The four small arrows represent the four consecutive A/D conversions that are performed at every time-step. The final output is given by the average of the four conversions. The dash-dotted line shows the 2.5 V offset added to the signal, corresponding to a digital zero after the A/D conversion.

5.3.3 The AO modulator arm

Figure 5.7 schematically illustrates how the AOMs are used in order to modulate the error signal $e(t)$ and, simultaneously, generate an intermediate frequency (IF). Two AOMs are used: the first AOM is driven by a signal $s_1(t) = A_1 e(t) \cos(\omega_1 t) + A_2 \cos(\omega_2 t)$, where A_1 and A_2 are RF carrier amplitudes, ω_1 and ω_2 are RF carrier angular frequencies. In our case, we have AOMs with RF center frequency $\omega_1/2\pi = 100.0$ MHz, RF bandwidth of 20 MHz (much larger than the signal bandwidth of 3kHz), illuminated with a spot size $d = 1.5$ mm. The photo-elastic medium is a TeO$_2$ crystal, with an acoustic wave velocity $v_a = 5$ mm/μs, providing us with a maximum modulated signal rise time $d/v_a = 0.3$ μs (much smaller than the minimum rise time of our error signal). $\omega_2/2\pi = 99.9$ MHz is offset from $\omega_1/2\pi$ by 100 kHz. Phase-locked synthesized signal generators were used in order to generate very stable RF carriers ω_1 and ω_2.

vert the digital output into a zero-balanced analog waveform. The LM6221 unit gain buffers [102] drive the 50 Ω lines used to drive RF amplifiers and the AO modulators [$E(k)$]. They also drive another 50 Ω line, $\overline{RW}$, which is set high during the read cycle, and low during the write cycle. This signal is provided through pin 6 of port A, and it is used to control the analog multiplexer in the differential amplifier circuitry (see Section 5.3.4) and to trigger supporting equipment used to monitor the experiment, such as oscilloscopes and spectrum analyzers. A picture of the actual digital control board is shown in figure 5.5. The M68HC11EVBU evaluation board is used [103]. The HC11 and its supporting circuitry are placed on the opposite side of the board. The coaxial connectors, input pin protection circuitry, unit gain buffers, op-amp, parallel connector and D/A converter are soldered to the vector board, as shown in the picture.

Figure 5.6 shows in more detail the output waveform and the timing of the analog-to-digital conversion. The SYNC signal, provided by the SLM driver, is shown as a dashed square wave. Whenever this signal is high, the SLM holds valid data. Whenever it is low, the driver is in the process of loading up the SLM with a new frame, and the SLM holds invalid data. For any given rate, the driver always takes 0.36 ms loading up a new frame. This means that, even though the driver can run up to 2 kHz, at rates higher than 1.5 kHz the duty cycle of the SYNC signal becomes severely limited, since the driver spends most of the time loading new data and very little time is left to use it for signal processing. At a sampling rate of 1.5 kHz, the SLM image is valid for about 0.31 ms, leaving enough time for heterodyne detection, amplification and digitalization. The solid line shows the output waveform sent to the digitizing port E. Soon after the rising edge is detected (98 ± 3 μs) the analog output is digitized four consecutive times and the result is stored in memory. The A/D results are summed and divided by 4 in order to provide an averaged readout, removing some of the high frequency noise.

Figure 5.7: Two AOMs are used to modulate the error signal and produce a SSB tone. An input beam is Bragg diffracted by the first AOM, producing an optic beam at its first order of diffraction modulated by the error signal $e(\kappa)$ at RF carrier frequency ω_1, plus a RF tone at ω_2. The second AOM diffracts the same beam at its -1 order, producing a final signal modulated by $e(\kappa)$ at RF baseband, plus a tone at the IF $\omega_1 - \omega_2$.

Their stability is important in our application, since the single-side-band (SSB) beat tone produced by mixing them together is used for active stabilization, explained in section 5.3.4. In the Bragg regime, the field diffracted by AOM1 is given by

$$E_a(x_o, t) \cong E_a\left(t - \frac{x_o}{v_a}\right) = a_2 G \sqrt{\eta_1}\, g\left(\frac{x_0 - a}{w_0}\right) E_0 e^{j(\omega t - kz)} \tilde{s}_1\left(t - \frac{x_o}{v_a}\right), \quad (5.4)$$

where $g(x) = \exp\{-x^2\}$ represents the Gaussian profile of an illuminating beam of width w_0 displaced by a, and $\tilde{s}_1(t)$ is the SSB analytic signal version of $s_1(t)$. Since the bandwidth of $e(t)$ is limited to a few kHz, the angular separation caused by its sidebands are negligible when compared to a 100 MHz RF carrier. Similarly, since ω_2 is so close to ω_1, the angular separation between these two tones is also unresolvable for the illuminated spot size. $a_2 E_o$ is the incident field amplitude, η_1 is the AOM amplitude diffraction efficiency, and G represents the electronic gain in the feedback. It is also assumed that the modulating signal power is small enough so that AOM1

operates in the linear regime. Another reason to keep its modulating power small is to prevent a large range in the variation of the power in the error signal. As the PR grating accumulates and the output signal gets stronger, the error signal becomes weaker, reducing the RF power modulating AOM1. If the initial power is too high, the resulting variation in temperature will be high enough to induce a change in the index of refraction of the material. The phase stabilization sub-system (section 5.3.5) can only handle a phase shift up to $\pm\pi$. To keep the total phase variation within range, it was empirically noticed that the maximum RF power at AOM1 should be limited to 50 mW.

The field E_a is Bragg matched with the first negative order of AOM2, driven by the single RF tone $s_2(t) = \cos(\omega_1 t)$. This provides a multiply diffracted order, which propagates collinearly to the original optical axis. It is passed through a pinhole spatial filter to completely block unwanted orders and to clean up the beam before reaching the PRC. It is then collimated, providing the following field at the PRC:

$$E^B(x_o, y_o, t) = C_a\, g(x_0, y_0) e^{j(\omega t - kz)}\left[A_1 e(t) + A_2 e^{-j(\omega_2 - \omega_1)t}\right], \quad (5.5)$$

where $C_a = a_1 G \sqrt{\eta_1 \eta_2}\, E_0$, and $g(x_0, y_0) = \exp[-(x_0^2 + y_0^2)/w_0]$ is the profile of the Gaussian beam. The final beam intensity at the PRC is measured to be 91 mW/cm². Note that the signal $e(t)$ is base-band modulated, as desired, while the tone at ω_2 is mixed with the carrier ω_1 in AOM2, generating a SSB tone at the IF $(\omega_2 - \omega_1)/2\pi =$ 100 kHz (chosen to be within the operating range of a digital lock-in amplifier), which is used for the active stabilization of phase drifts in the interferometric detection system. Note that the IF tone at 100 kHz is far from the frequency range of the signals at the SLM (ranging in frequency up to ≈ 3 kHz). As such, the IF tone does not record a coherent grating with the SLM beam at the PR crystal. Also $A_1 \gg A_2$, and only the error modulation is relevant to signal processing. Thus, in discrete notation, we have

$$E^B(\kappa) = C_a' e(\kappa), \quad (5.6)$$
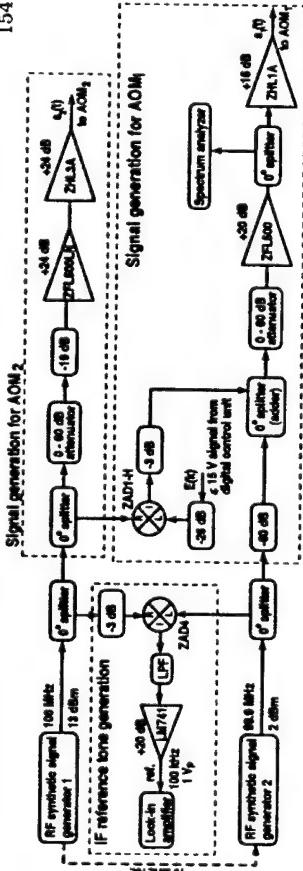
Figure 5.8: RF circuitry used for generating signals used to modulate AOM_1, AOM_2, and to provide the heterodyne stabilizing lock-in amplifier with a reference IF tone.

where $C'_a = C_a g(x_0, y_0)$.

The diagram in figure 5.8 shows how the signals $s_1(t)$ and $s_2(t)$ are generated. It also describes the generation of the IF reference tone used by the lock-in amplifier in the heterodyne stabilization system (see Section 5.3.5). The two synthesized RF generators are locked through their 10 MHz internal clocks, providing long-term stability to the IF tone. RF modules provided by Mini-Circuits [104] have been used to implement each one of the splitters, attenuators, mixers and amplifiers shown in the diagram. The signal from the first signal generator is first split and then attenuated by 3 dB in order to bring it within the optimum range of operation of the ZAD4 mixer, where it is mixed with the signal from the second signal generator. The resulting signal is low-pass filtered in order to remove the sum-of-frequencies tone. An LM741 op-amp in the inverting configuration is used to bring the filtered signal within the optimum range of operation of the reference tone used by the lock-in amplifier.

Signal generation for AOM_2 is performed as follows: a variable attenuator is used in order to control the total signal power going to the AOM. It is capable of attenuation ranging from zero to 60 dB, but in practice about 6 dB of attenuation is used. A fixed 19 dB attenuator is used to prevent damage to the first amplifier, a low-noise ZFL500LN. The high-power amplifier, a ZHL3A, is selected such that it can provide a large amount

of gain (24 dB), but with a maximum output power of 29.5 dBm, below the maximum AOM input power.

Signal generation for AOM_1 is performed as follows: signal $E(k)$ from the digital control unit is attenuated by 26 dB, assuring its power level to be within the operational range of the ZAD1-H, a mixer with high LO-RF isolation (30 dB min.). This high level of isolation is critical in order to suppress as much of the RF carrier in the modulated signal as possible, since any remaining RF carrier will translate into a DC signal during photorefractive recording, contributing to the decay of the PR grating and writing a grating with any leakage from the DC spatial filter in the SLM arm. The modulated signal is added to the carrier from the second signal generator, which has been strongly attenuated (60 dB). The strong attenuation is present in order to prevent the optical IF tone from affecting the PR grating recording. The attenuated IF signal, however, is easily detected by the high-gain lock-in amplifier. The combined signals are followed by a variable attenuator, allowing for the control of the power of the signal sent to the AOM. In general this attenuator is set at -10 dB in order to prevent non-linear modulation at the AOM_1. After the first amplifier (a low-noise ZFL500) the signal is split, so that it can be monitored at a spectrum analyzer. The remaining signal is amplified by the ZHL1A high power amplifier. Again, proper care is taken to prevent damage to the AOM by selecting an amplifier with maximum output power below the maximum allowable input of the AOM.

5.3.4 Differential heterodyne detection

In our setup we observed a strong low frequency background noise associated with fluctuations in the intensity of our Ar^+ laser ($\nu = 514$ nm), known as relative intensity noise (RIN). Figure 5.9 shows a technique which allows us to eliminate such common mode laser noise that is common to both interferometric arms. Energy conservation at the dielectric interface guarantees that the signals detected by detectors 1 and 2 must be

where the undesired terms $|E_0|^2 m(t)|^2$ and $|E_0|^2$ have been canceled, leaving aside the desired term $2m(t)|E_{av}|^2$ and the remaining noise term $4m(t)E_{av}E_n^*(t) + c.c.$, which is small compared to the desired term for a large $|E_{av}|^2$.

Heterodyne detection requires a constant phase difference between the reference and the signal beam. Various sources, such as mechanical vibrations, air currents, temperature changes, and micro-creep of the optic mounts produce phase drift. To compensate for such drifts, passive stabilization was used, which included enclosing the entire optic system. Such measures limited the phase drift to about π radians over a time period of half an hour. The effect of the residual phase drift depends on whether the detected signal is SSB or double side band (DSB) modulated. For a DSB signal, the field amplitude of the signal beam is represented by

$$E_s^D(t) = \frac{E_s}{2}\cos(w_s t)e^{i(\omega t + \phi_s)} + c.c.,$$ (5.8)

where w_s is the angular frequency of the signal of interest and ϕ_s is its initial optical phase. After interfering with a reference beam $E_r(t) = \frac{E_r}{2}e^{i(\omega t+\phi_r)}+c.c.$, the heterodyne signal of interest is given by

$$I_h(t) = E_s^D(t)E_r(t)^* + c.c. = E_s E_r \cos(w_s t)\cos(\phi_s - \phi_r),$$ (5.9)

that is, the signal amplitude is modulated by the cosine of the interferometer phase difference $\phi_s - \phi_r$ and, therefore, this difference must be kept close to zero. A SSB signal, on the other hand, is represented by

$$E_s^S(t) = \frac{E_s}{2}e^{iw_s t}e^{i(\omega t+\phi_s)} + c.c.$$ (5.10)

and the heterodyne signal of interest becomes

$$I_h(t) = \frac{E_s E_r}{2}\cos(w_s t + \phi_s - \phi_r).$$ (5.11)

That is, the phase difference causes a phase shift of the detected signal, but it does not affect its amplitude.

Figure 5.9: Differential heterodyning and long term interferometric stabilization. The signals detected by detectors 1 and 2 are subtracted by the differential amplifier, removing common mode noise and increasing the dynamic range of the detected signal. The output signal is fed to the lock-in amplifier, which determines the phase of the reference tone by comparing it to an electronic lock-in reference. Phase drifts are corrected by moving a mirror attached to a piezo-electric actuator.

π radians out of phase. Therefore, it is possible to eliminate DC and common mode RIN by subtracting the detector outputs [105]. This operation is performed by a differential amplifier. Let the field amplitude of the laser be given by $E_0 = E_{av} + E_n(t)$, where E_{av} is the average field amplitude and $E_n(t)$ is the noise amplitude. For a signal modulation $m(t)E_0$ of one arm of the interferometer, the voltage output at each detector is given by $V_1 \propto |E_{av} + E_n(t)|^2|m(t) + 1|^2$ and $V_2 \propto |E_{av} + E_n(t)|^2|m(t) - 1|^2$. The difference between these two voltages gives us

$$V_{out} = V_1 - V_2 \propto 2m(t)[|E_{av}|^2 + 2E_{av}E_n^*(t) + c.c.]$$ (5.7)

is used to quickly select between two different input ports, with the minimum possible switching delay. The multiplexer uses the RW signal, generated by the micro-controller, in order to select the signal sent to the amplifier in the next stage. During the read cycle (RW set high) the output of the second stage is passed on to the fourth stage. During the write cycle (RW set low) a logic ground is passed to the next two stages.

The fourth stage consists of an OPA37 [108] variable gain amplifier in the inverting configuration, capable of a voltage gain ranging from -27 dB up to 28 dB. This provides the operator with enough flexibility to control to overall processor gain by varying a single parameter. Note that decoupling capacitors have been purposely omitted. This is the case because their time constant would be inevitably inside the time transient of the broadband temporally-multiplexed output signal. Instead, offset canceling trim-pots are provided at each op-amp.

The final stage consists of an OPA227 used as an adder, adding a constant 2.5 V offset to the final output. This is required since the A/D converter built into the micro-controller operates in the voltage range of 0 V to 5 V. This stage also provides extra protection to the A/D converter, preventing the output voltage from exceeding the A/D input limit by selecting adequate supply voltages, saturating the op-amp whenever the limits are approached.

5.3.5 Active stabilization in detection

As explained in section 5.3.3, the AOM arm of the optical processor is DSB modulated by the signal of interest plus a 100 kHz SSB tone. Initially, the system is aligned in order to provide maximum amplitude at the DSB modulated signal, at which point the phase difference $\phi_s - \phi_r$ equals to zero. Passive stabilization guarantees that this signal will be stable for a few seconds and provides enough time to initiate active stabilization. Active heterodyne stabilization is performed using a SR850 DSP Lock-in amplifier in combination with a mirror attached to a piezo-electric actuator, as shown in

---

Stage 1 — Differential amplifier
Stage 2 — High gain amplifier
Stage 3 — Analog multiplexer
Stage 4 — Variable gain amplifier
Stage 5 — Offset adder

Figure 5.10: Differential detection circuitry. First stage: differential detector. Second stage: high gain amplifier used to bring the signal levels up to the optimum range for multiplexer operation. Third stage: high speed analog multiplexer, used to prevent the last two stages from saturating. Fourth stage: variable gain multiplexer. Fifth stage: offset adder, used to bring the signal within the mid-level of the A/D converter.

The diagram in figure 5.10 depicts the electronic circuit used in differential detection, amplification and temporal multiplexing. The first stage consists of an OPA227 [106] low-noise, high common mode rejection ratio (CMRR) op-amp used as a low-gain differential amplifier. Two trim-pots are used in series with the input resistors. They provide the operator with the freedom necessary to balance the signals at each input port, thus enabling the removal of the large DC bias resulting from heterodyne detection. The output of the first stage is directly fed to the lock-in amplifier, represented by signal V_{out} in figure 5.9. The second stage consists of a low-noise OPA228 used as an inverting amplifier with 20 dB of gain. It is used in order to bring the signal voltage range to about ±2 V, the optimum input voltage to the analog multiplexer used in the third stage. During the write cycle the detectors are flooded by light of high intensity from the AO arm. Consequently, the signals in the differential amplifier circuitry have much higher power than in the read cycle. This signal is not high enough to saturate the first two stages, but it is high enough to saturate the last two. Once saturated, op-amps require some time to return to normal operation, and such a delay translates into a degradation of the temporally multiplexed signal at the beginning of every read cycle. In order to prevent this degradation, the CLC532 [107] high speed analog multiplexer

Figure 5.11: Photorefractive anti-domain visualization by imaging the bright regions of a PRC in a two-wave-mixing experiment in which the crystal axis is counter-propagating with respect to the pump beam. The inset shows the boundary between an anti-domain and a regular domain. While the pump loses intensity over the majority of the crystal, it gains intensity in the anti-domain.

suitable for use with the FIRONN architecture. This architecture requires the recording of two-dimensional gratings such that the selected crystal must present a considerable region with a minimum amount of anti-domains. Figure 5.11 shows the experimental setup used to map the anti-domains. A collimated beam is split in two and the two resulting beams are set to interfere at the PR crystal with the pump beam propagating opposite to the crystal c-axis. A lens with long focal length images the crystal onto a CCD camera, such that the resulting imaging system has a depth of focus relatively high compared to the length of the crystal. The two beams record a grating and two-wave mixing takes place [113], transferring power from the pump beam to the signal beam, such that the image of the crystal becomes progressively darker. If 180° domains (or

figure 5.9. The RF carriers are mixed and the resulting 100 kHz difference frequency is used as a reference by the lock-in amplifier, which detects the phase of the SSB detected signal. Because of the high gain and narrow filtering provided by the lock-in, this tone can be set at 40 dB below the DSB modulated signal. The low amplitude of the SSB tone, its high frequency, and its lack of correlation with the SLM signals prevents it from interfering with the processor adaptation. The time constant of the lock-in amplifier is set at 30 ms, which is fast enough to compensate for most of the residual phase drift, and is slow enough to provide the lock-in with the gain and filtering necessary to reliably detect the 100 kHz tone. The phase output of the digital lock-in amplifier, proportional to $\phi_s - \phi_r$, is then connected to a high-voltage amplifier through a voltage divider, which gives us control of the gain. This amplifier drives the piezo-electric translator, moving the mirror back and forth in the reference arm of the heterodyne detector, correcting for any drifts in the phase.

Note that such a system stabilizes only the heterodyne detection process and does not affect the holographic grating formation in the PR crystal. However, for the slow phase drifts in the system, the crystal is capable of adaptively forming new gratings whenever the SLM arm and the AOM arm experience a phase change. The heterodyne detector, on the other hand, would cease to work whenever its phase difference reached π radians, and it would remain in that state for a time period over several minutes. That is the reason why active stabilization is so important in heterodyne detection, although it is not mandatory for PR grating formation.

5.3.6   180° domains visualization and repoling of the SBN crystal

Preliminary experiments were performed using a 0°-cut, [10×11×5]mm³ Cu-doped strontium barium niobate crystal (SBN:75), of tetragonal point group $4mm$ [109]. This crystal, however, will not be described in detail, since it had to be replaced due to the presence of a large concentration of 180° domains [110, 111, 112], rendering it un-

Figure 5.12: Anti-domain visualization in a SBN crystal before the repoling attempt, showing a high concentration of 180° domains.

Figure 5.13: Anti-domain visualization after the repoling attempt, showing almost no modification in the 180° domain distribution.

electro-optic tensor with its coefficients is given by [49]

$$\begin{bmatrix} 0 & 0 & r_{13} \\ 0 & 0 & r_{13} \\ 0 & 0 & r_{33} \\ 0 & r_{42} & 0 \\ r_{42} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5.12}$$

where $r_{42} = r_{51}$ is the dominant electro-optic coefficient for BaTiO$_3$. At $\lambda = 546$ nm, $r_{42} = 1640$. The material has an ordinary index of refraction $n_o = 2.437$, and an extraordinary index of refraction $n_e = 2.365$.

Figure 5.14 shows a map of the birefringence of the BaTiO$_3$ crystal. It is done by imaging the crystal while illuminated by a single circularly polarized beam propagating normally to the crystal surface, followed by an analyzer placed after the crystal. With a perfectly collimated beam, regions with the same amount of birefringence change the polarization of the incident beam by the same amount and, therefore, present equal

anti-domains) are present they determine the transfer of power to take place from the signal beam towards the pump beam, and the anti-domain regions become brightly illuminated. Figure 5.12 shows the resulting picture for the SBN crystal, clearly showing the presence of multiple 180° domains. An attempt was made to remove these anti-domains from the crystal by electrically repoling it. First its Curie temperature was determined by measuring its DC capacitance as its temperature was increased in an electric oven. It was found that $T_c = 55 \pm 2°C$. Next, the sign of the majority charge carriers was determined by measuring the direction of the electric field due to the pyro-electric effect when the crystal is heated. Electrons were determined to be the majority carriers, meaning that the field applied to the crystal during repoling has to be in the same direction as the crystal c-axis. The following repoling schedule was used: the temperature was linearly increased from room temperature (27.5°C) to 45°C in ten minutes. Then it was increased to 70°C (clearly above T_c) in 25 minutes, when a field of 4 kV/cm was applied for 30 minutes. The field was removed and the crystal was slowly cooled down to 35°C in 35 minutes, when it was left to naturally cool back to room temperature.

Figure 5.13 shows the 180° domains after repoling, clearly showing that repoling did not result in any improvement. The final temperature, electric field and repoling time used were slightly higher than those suggested in the literature. Thus, it was concluded that it would again be fruitless and perhaps imprudent to insist on a more aggressive repoling schedule. Instead, a new crystal was selected.

5.3.7 The BaTiO$_3$ crystal

All the following experimental results reported on this thesis have been obtained using a 45°-cut ferroelectric Fe-doped barium titanate (BaTiO$_3$) crystal of dimensions [5×5×3]mm^3, which also has a 4mm point group symmetry at room temperature. The

Figure 5.14: Birefringence visualization in the BaTiO$_3$, showing a large uniform region at the center of the crystal.

projections in the axes of polarization of the analyzer. Thus, they are seen as regions of equal intensity. However, even a well collimated beam will still present some amount of divergence. In a 45°-cut crystal this will result in different optical path differences for the ordinary and extraordinary components of the polarization. Consequently, some fringes will be present even when the crystal is uniform, but one can still infer that regions without many fringes represent regions where the crystal is mostly uniform, and regions with multiple, closely spaced fringes represent regions where the crystal birefringence is changing rapidly and may be a consequence of internal stress or imperfections in the crystal structure. Note the presence of a large uniform region in the center of the crystal. Also note the presence of a strong imperfection close to the upper right corner of the crystal. Cleaning the crystal with a spray of purified air caused it to cool too
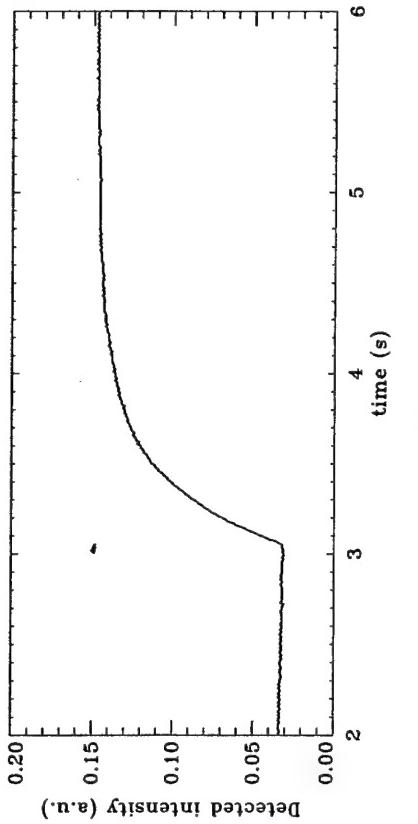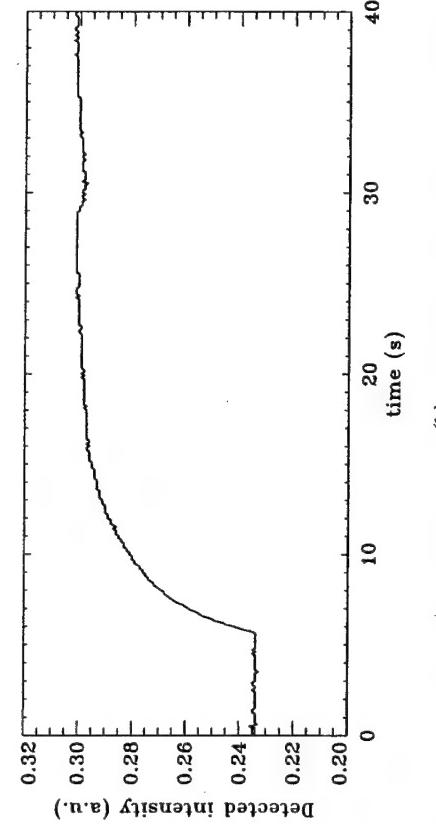
Figure 5.15: Signal beam intensity in two-wave-mixing with the BaTiO$_3$ crystal.(a) — total intensity = 455 mW/cm^2, time response = 0.39 s. (b) — total intensity = 45 mW/cm^2, time response = 4.73 s.

rapidly and caused physical damage (a small crack) on that corner. Consequently, the central region, away from the crack, is selected as the best region to record dynamic holograms during experiments.

Figure 5.15 depicts the time evolution of the intensity of the signal beam in a two-wave mixing experiment with the BaTiO$_3$ crystal. In figure 5.15(a) the power of

Figure 5.16: The two-wave mixing geometry used during adaptation. The beam ratio and the orientation of the crystal axis were chosen in order to maximize the power of the diffracted beam, while minimizing the power of PR beam fanning propagating towards the output detectors (along the AOM beam).

amplitude with time and, consequently, the power in the AOM beam tends to decrease, increasing the intensity ratio between the beams. However, the total intensity incident at the crystal does not change much, having a lower limit of 3.65 mW/cm² (the intensity of the SLM beam). This limited variation in intensity, however, is beneficial to the adaptation, since the variation of the PR time response with the optical power in the error beam makes the processor adapt quickly initially, and slower once it gets closer to the final solution, reducing the amount of misadjustment. This behavior is similar to that suggested by the normalized LMS algorithm, which sets the learning rate to a value proportional to the power of the error signal [114], promoting quick learning at first, and small misadjustment at the end of adaptation.

5.3.8 Output formation by PR diffraction and spatial integration

The photorefractive grating formation is now analyzed using a simplified model of the dynamics of the PR crystal, i.e., assuming diffusion dominated charge transport and ignoring the effect of beam coupling on the diffraction efficiency. Under these assumptions, the space-time evolution of the refractive index modulation $\Delta n(x, y, z, t)$

the CW incident beams is 2.5 mW for the pump beam and 0.25 mW for the signal beam (pump to signal ratio $r = 10$). They overlap over an area of 7.7 mm², with a total intensity of 35.7 mW/cm², chosen by maximizing the steady-state diffraction efficiency. The signal beam is initially blocked, and it is unblocked at $t_0 = 3.06$ sec. At $t_1 = 4.23$ sec the intensity of the signal beam after the crystal reaches 95% of its value at steady state [$.950 = 1 - \exp(-3)$]. Estimating the PR time constant as $\tau_{pr} = \frac{t_1-t_0}{3}$, we find that $\tau_{pr}(I) = 0.39$ sec under these conditions. Figure 5.15(b) shows the same experiment, but with a pump beam power of 220 μW and a signal beam power of 22 μW ($r = 10$). The total beam intensity is 3.14 mW. Under these conditions, we have that $t_0 = 5.5$ sec, $t_1 = 19.8$ sec, and $\tau_{pr}(I) = 4.73$ sec. Note that the PR time constant can change quite dramatically with a reduction in the intensity of the interfering beams.

Figure 5.16 depicts the two-wave mixing geometry used during adaptation. The power of the SLM beam is 0.28 mW. The power of the AOM beam after the square aperture is 250 μW. The aperture is 5mm×5.5mm, and it is set so that it exactly matches the square image diffracted by the SLM beam when the AOM beam is blocked. Measuring the power of the AOM beam after the aperture assures us that we are measuring only the power of the portion of that beam which overlaps with the SLM beam and contributes to grating formation. The intensity of the SLM beam is 3.64 mW/cm² and that of the AOM beam after the square aperture is 0.91 mW/cm² for an intensity ratio of 4. Our goal is not to maximize the small signal gain, but to maximize the power diffracted by the SLM beam during holographic readout. That is the reason why the SLM beam is set to be stronger than the AOM beam. Also note that the crystal axis is oriented such that the PR beam fanning [63] emanating from the SLM beam is directed away from the heterodyne detectors, preventing fanning from interfering with the detection of the output signal.

During closed-loop adaptation the error feedback signal beam tends to decrease in

in the PR medium is given by equation 1.31, repeated below for our signals of interest:

$$\Delta n(x,y,z;t) = \beta e^{\frac{-t}{\tau_{pr}}} \int_0^t E^A(x,y,z;t') E^{B*}(x,y,z;t') e^{\frac{t'}{\tau_{pr}}} dt', \qquad (5.13)$$

where E^A and E^B are the space and time-varying amplitude distributions of the writing beams coming from the SLM and AOM arms, respectively (Eqns. 5.3 and 5.5). β is the sensitivity of the PR crystal (*i.e.*, the change in the index of refraction per unit energy per unit area), and τ_{pr} is the intensity dependent time response of the crystal.

Representing E^A and E^B in discrete notation, taking in consideration the separation between read and write cycles used in temporal multiplexing, we get to the following expression

$$\Delta n_{i,\tau}(\kappa) = C\gamma^\kappa \sum_{p=0}^{P/2} \sum_{\kappa'=0}^{T} e^*\left((2p-1)(T+1)+\kappa\right) s_i(\kappa'-\tau)\gamma^{-1}, \qquad (5.14)$$

where $C = \beta C_s C_a'$, $\gamma = e^{-\frac{\Delta_t}{\tau_{pr}}}$ is a factor used to account for the PR decay and grating accumulation. For successful adaptation the time decay should be slow and, therefore, γ should be close to 1. That is, the PR time constant τ_{pr} should be much larger than the time step Δ_t, a condition which is readily met in our system ($\gamma \simeq .99968$). $P = \kappa$ DIV $(T+1)$ is a counter for the current cycle (read or write). By definition, the system starts in a read cycle (when $P = 0$) and P is even during read cycles and odd at write cycles. Note that the error term $e^*((2p-1)(T+1)+\kappa')$ recalls the error detected in the previous read cycle $(2p-1)$, with periodicity $2p(T+1)$. Also note that equation 5.14 is only valid during read cycles. However, that is not a problem, since we are interested in the final output, which is only produced during read cycles anyway. Finally, notice that the periodicity of the read cycle has been dropped out in the term $s(\kappa'-\tau)$. This is possible thanks to the periodicity of the input signal represented by the limited number of frames $(T+1)$ in the SLM driver, such that $s(\kappa - p(T+1)) = s(\kappa)$ for any integer p.

Since $\Delta n_{i,\tau}(\kappa)$ is expressed only during read cycles (when all the recording has taken place over previous cycles) it is possible to take the term γ out of the summations

---

and rewrite the constant at the front. Also, assuming T to be large such that $T+1 \simeq T$, we are capable of simplifying the resulting expression, giving us

$$\Delta n_{i,\tau}(\kappa) = C'\kappa\gamma^{\kappa-1} \sum_{p=0}^{P/2} \sum_{\kappa'=0}^{T} e^*\left((2p-1)(T+1)+\kappa'\right) s_i(\kappa'-\tau), \qquad (5.15)$$

where $C' = C/2$.

The output signal is given by the Bragg matched diffractive readout of the dynamic grating with the SLM beam, followed by spatial integration at the detectors. Using discrete notation, the output signal after differential heterodyne detection is given by

$$o(\kappa) = \eta_d Z_0 g_e \sum_{i=1}^{N} \sum_{\tau=0}^{T} s_i(\kappa-\tau)\, \Delta n_{i,\tau}(\kappa), \qquad (5.16)$$

which is valid when κ DIV $(T+1)$ is even (read cycle), and where η_d is the responsivity of the detectors, Z_0 is the impedance of the transimpedance amplifiers, and g_e is the electronic gain that follows detection. Substituting $\Delta n_{i,\tau}(\kappa)$ using equation 5.15, we have

$$o(\kappa) = C''\kappa\gamma^{\kappa-1} \sum_{i=1}^{N} \sum_{\tau=0}^{T} s_i(\kappa-\tau) \sum_{p=0}^{P/2} \sum_{\kappa'=0}^{T} e^*\left((2p-1)(T+1)+\kappa'\right) s_i(\kappa'-\tau), \qquad (5.17)$$

where $C'' = C'\eta_d Z_0 g_e$. Note that equation 5.17 is in the same form as the correlation-cancellation loop operation required for jammer nulling in adaptive antenna arrays (Eqn. 1.24).

5.4 Experimental results

The experimental setup has the advantage of easily allowing the change of the input signal scenarios, thus providing good flexibility to test the performance of the processor under different conditions. The BTSIM program (see appendix B) is used to generate the input signal data. Another program is used to convert it into the 128 individual frames under a format which can be understood by the SLM driver. Some of the input signal scenarios investigated are: narrowband or broadband beam-forming
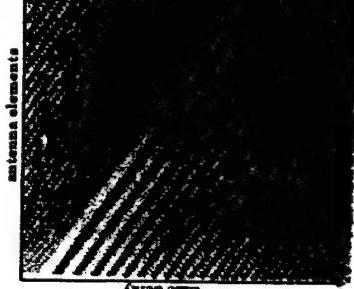
Figure 5.17: One of the 128 images captured from the SLM, showing a broadband binary chirp in the far-field, when the SLM operates as an amplitude modulator.
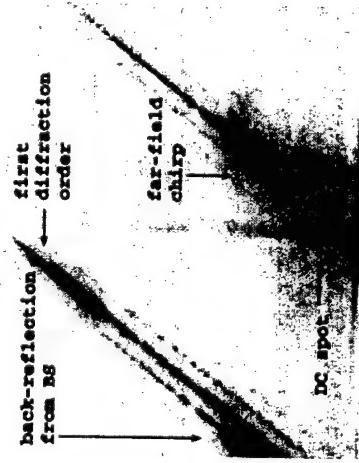
Figure 5.18: Spatial-temporal spectrum of the array environment displayed in the optical Fourier plane of the SLM, showing a thin and straight line, characteristic of a broadband signal in the far-field. The DC spot and multiple diffraction orders are also visible.

and jammer nulling, far-field and near-field signals, linear or circular arrays, multiple signals and jammers, and any combination of the above. Simulations of the following scenarios were presented in chapter 4 for comparison with the experiments presented here. They offer a good representation of the overall performance of the array processor.

5.4.1 Far-field beam-forming

In this first, simple scenario, the processor performs beam-forming in the direction of a broadband signal source (a linear, binary chirp, generated from the binary quantization of an analog chirp ranging in frequency from 168.75 Hz to 518.75 Hz) located in the far-field of the antenna array (AOA = 0.25 radians). Figure 5.17 shows one of the frames displayed on the SLM (amplitude modulation has been used to display this image, but the SLM normally operates as a phase modulator). The chirp wave-fronts are straight, characteristic of a signal in the far-field, and they are tilted with an angle proportional to the sine of the AOA of the corresponding sonar signal being detected.

Figure 5.18 shows an image captured at the Fourier plane of the SLM, where the spectrum of the broadband chirp is represented as a long and thin line, tilted at an angle perpendicular to the chirps seen in figure 5.17. The line is long because of the multiple frequencies present in the chirp. The lowest frequencies occupy the positions closest to the DC spot, while the highest frequencies occupy the positions farthest away from the DC spot. Such a spectrum is characteristic of a broadband signal which, therefore, requires true-time-delay processing in order to prevent beam-squint (See section 1.3.4 for more details on beam-squint). The line is long, as expected from a broadband signal, and thin, implying that the signal is coming from a specific AOA and, therefore, is present in the far-field of the antenna array. A strong DC spot and multiple orders of diffraction are also visible. The back-reflection of the beam-splitter are only present at the 2-f system used to visualize the the SLM in the Fourier plane. Thus, they are not present at the PR crystal. The multiple diffraction orders and the DC spot, on the other hand, are spatially filtered before reaching the PRC.

Figure 5.19 shows the desired signal $d(\kappa)$ used in the feedback loop represented in the time domain [(a)] and in the frequency domain [(b)]. Note that the non-linear binary quantization and the temporal windowing of the signal caused it to spread over frequencies beyond the originally selected bandwidth. Once the SLM starts scrolling and the PRC starts correlating its image with the feedback error signal, the first results of the adaptation are already seen after less than 1 second of processing time. It takes about 3 seconds until the adaptation reaches its steady state. Figure 5.20(a) shows a typical waveform of the detected output at this stage, acquired through an oscilloscope and then downloaded to a computer through a data acquisition card. In order to better compare it with the desired signal, this signal is thresholded about zero [Fig. 5.20(b)], giving us back the desired signal, as expected. Its bit error rate (BER) has been calculated by comparing the output bit sequence with the bit sequence of the input after interpolating it in order to provide it with as many samples as the output waveform acquired from
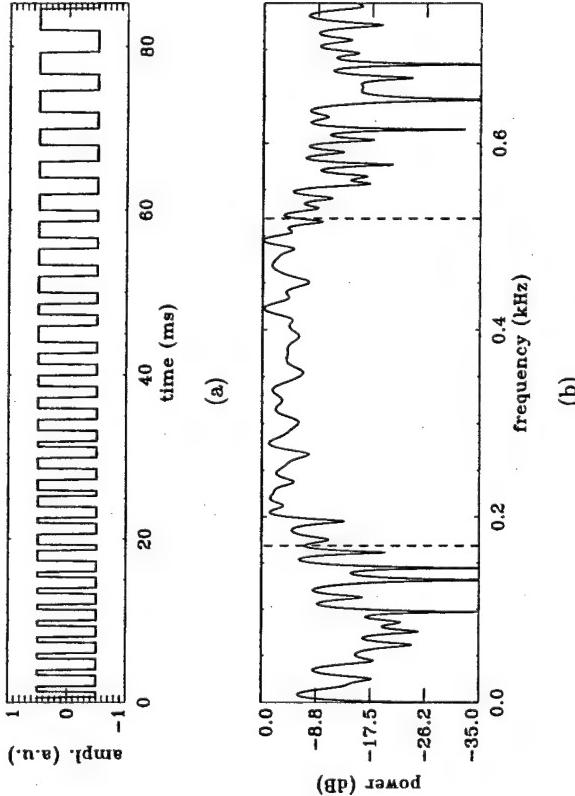
Figure 5.19: (a) Binary chirp used as desired signal. (b) Fourier spectrum of the desired signal. The dashed lines depict the nominal bandwidth of the analog chirp from which the binary chirp was derived.

the digital oscilloscope. This method, although not very accurate, provides us with a quantitative mean to compare the performance of the array under different scenarios. The BER in this case is 5.93×10^{-3}.

Figure 5.20(c) shows a plot of the spectrum of the detected output to be compared with the spectrum shown in figure 5.19(b). Note that the Nyquist frequency in this case is 0.75 kHz, and the output spectrum is very close to the spectrum of the desired signal, and it is about 10 dB over the background noise. The dashed lines depict the initial and final frequencies of the original analog chirp used to generate the desired binary chirp.

The experimental setup allows the display of a pattern proportional to the amplitude of the adaptive weights, as shown in figure 5.21. This image is produced by the diffraction of light from the AOM arm by the dynamic holographic gratings, followed
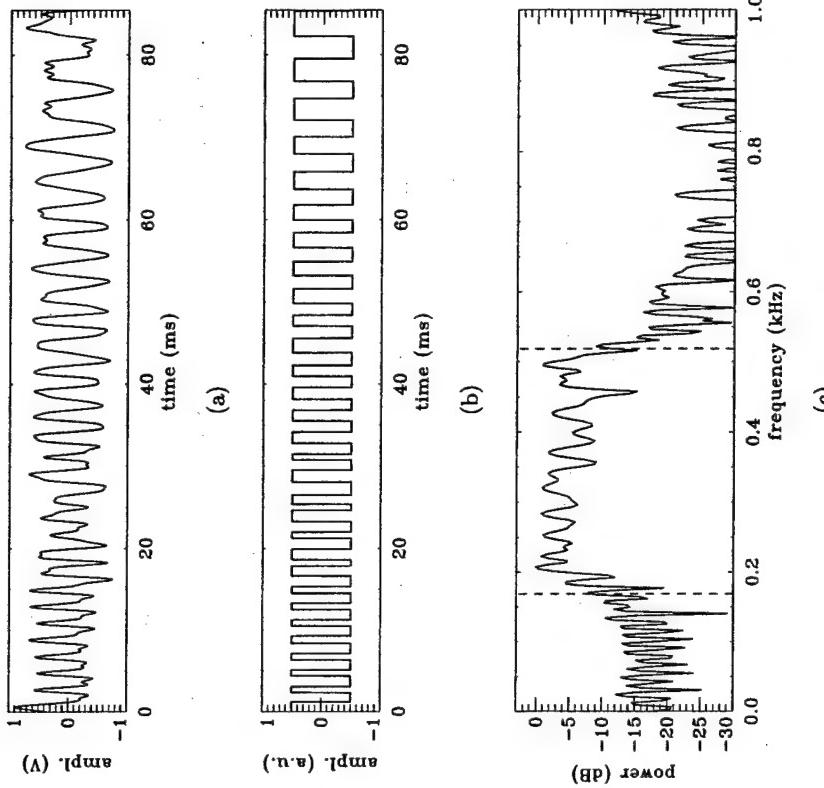
Figure 5.20: (a) Processor output before thresholding, showing that the broadband chirp is fully recovered. (b) Processor output after thresholding, showing that the binary desired signal is fully recovered. (c) Output spectrum, showing the detection of a broadband signal in the frequency range of the desired chirp. The dashed lines depict the initial and final frequencies of the original desired chirp.

by a time-integrated square-modulus detection at a CCD camera (CCD_3 in Fig. 5.2, placed in an image plane of the SLM). The diffracted image intensity is proportional to the strength of the PR gratings and, therefore, proportional to the magnitude of the adaptive weights they represent. As seen in chapter 1 (section 1.1.5, Eqn. 1.23), the adaptive weights converge to a multi-channel correlation between the desired signal and

antenna elements i

time delays τ

Figure 5.21: SLM image diffracted from the PRC, showing the correlation pattern of the far-field desired chirp.
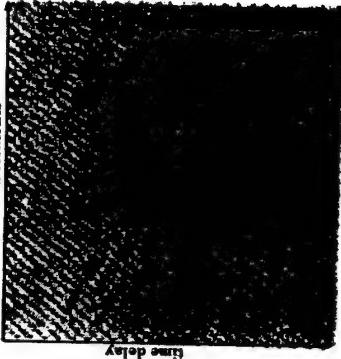
Figure 5.22: Correlation pattern in the Fourier plane of the SLM, showing that the systems responds to a broadband signal in the far-field.

the feedback error signal which, when only the desired signal is present, corresponds to the auto-correlation of the desired signal. Physically, this correlation is performed by the time integration of the PR crystal as it accumulates the interference grating between the SLM and AOM beams. In the case of a broadband signal in the far-field the correlation is a thin and straight line, where the tilt of the line corresponds to the amount of time delay necessary to coherently add the signal from each antenna element in order to reconstruct a combined output signal (see section 5.5 for more details). Many of the correlation side-lobes are also visible. Compare this figure to figure 4.43 in chapter 4 and note the similarity between this pattern and the weights after adaptation in the far-field binary chirp simulation.

Figure 5.22 shows an image obtained at the Fourier plane of the SLM of the light diffracted from the AOM arm by the dynamic holographic gratings, detected at CCD_4 in figure 5.2. It is an image of the Fourier transform of the correlation pattern and, therefore, it is proportional to the magnitude of the cross-power spectral density

antenna elements

time delay

Figure 5.23: One of the 128 images captured from the SLM, showing a broadband chirp in the near-field and a CW jammer in the far-field.

Figure 5.24: Image of the SLM in the Fourier plane, showing a bright dot (the CW jammer) and a circular section, characteristic of a broadband signal in the near-field.

between the input signal and the error signal. It is also proportional to the Fourier transform of the weights.

5.4.2   Near-field beam-forming and CW jammer nulling

In this second scenario, the processor performs beam-forming in the direction of a broadband chirp in the same frequency range as the one in the previous experiment, but this time located in the near-field of the antenna array, at a distance $4L$ from the plane containing the array, and $1.2L$ from the normal to the array, where L is the physical size of the antenna array (see Fig. 4.46). Simultaneously, the processor nulls out a narrowband chirp ($f_0 = 281$ Hz) present on the far-field (AOA = -0.5 radians) with twice as much total power as the desired signal. Figure 5.23 shows one of the frames displayed on the SLM, where the curvature of the near-field chirp signal can be noticed. Figure 5.24 shows the corresponding Fourier plane of the SLM, where the two signals can be easily distinguished. The CW jammer, represented by a single frequency at a particular angle of arrival, is seen as a bright dot, since all its energy is concentrated

at a single point in the signal spectrum. The near-field broadband chirp, on the other hand, has its energy spread over a range of spatial and temporal frequencies seen over a circular section. The binary quantization, which is a non-linear process, results in the mixing of the positive and negative side lobes of the chirp spectrum with the jammer spectrum, manifested as the symmetric duplication of the signal spectrum centered at the jammer's second harmonic, but with lower intensity. The simulations presented in chapter 4 showed that such non-linear effects could be handled by the processor without hindering adaptation, and experimental results confirm the simulations.

As in chapter 4, two cases are considered: open-loop and closed-loop operation.

In open-loop, the detected output $o(\kappa)$ is not subtracted from the desired signal, such that $e(\kappa) = d(\kappa)$ and a conventional correlation between the input and desired signal is calculated, instead of an adaptive correlation-cancellation loop. Figure 5.25(a) shows the thresholded output after steady state for this case. Note that only the CW jammer is detected, as evidenced in figure 5.25(b), depicting the spectrum of the output signal. In this mode of operation, although the PR gratings converges to a correlation between the desired signal and the input chirp, a considerable amount of the CW jammer is diffracted by the same correlation gratings. Since the jammer has all of its power concentrated at a single frequency, this results in an output with much higher power at the jammer frequency than at any other signal frequency. In a real-world scenario, however, it is unlikely that the jammer would be coherent with the desired signal, and the resulting correlation gratings would wash out.

In closed-loop, the processor returns to operate as a correlation-cancellation loop, subtracting the output signal from the desired signal such that the feedback error signal is given by $e(\kappa) = d(\kappa) - o(\kappa)$ and performs a correlation-cancellation loop. That is, any jammer power diffracted from the PR grating and detected at the output will be fed back 180° out of phase. This feedback signal will promptly write a grating with the exact phase necessary to null out the jammer signal present in the input, but with the exact phase necessary to null out the
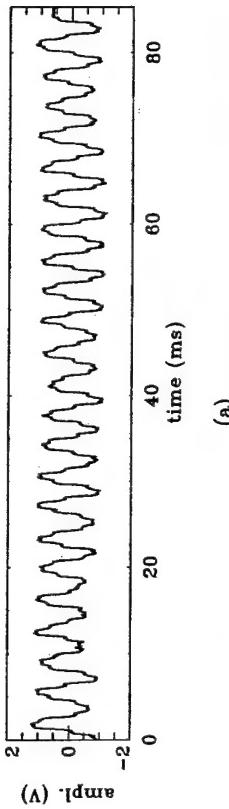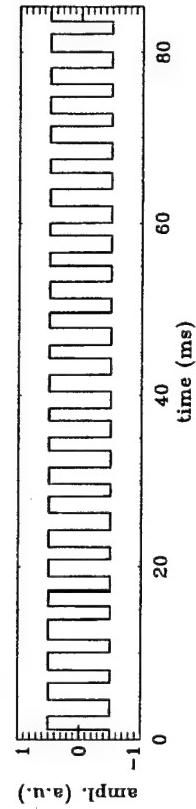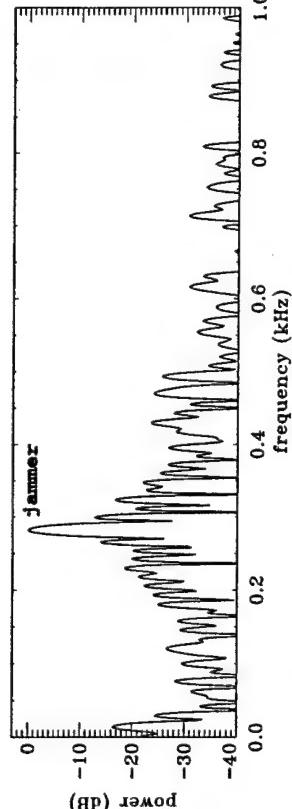
Figure 5.25: (a) Open-loop processor output before thresholding, showing that only the CW jammer is detected. (b) Open-loop processor output after thresholding. (c) Open-loop processor output in the frequency domain, showing a strong peak at the jammer frequency (281 Hz).

jammer. This results in a dramatic difference in performance, as seen in figure 5.25(c). This time the desired broadband chirp is detected, albeit not as cleanly as in the case of simple beam-forming [Fig. 5.20(a)]. The output signal spectrum [Fig. 5.25(d)] clearly shows that the jammer has been reduced, and that the signal detected occupies the same bandwidth as the desired chirp. The BER achieved for this case is 5.99×10^{-3}.
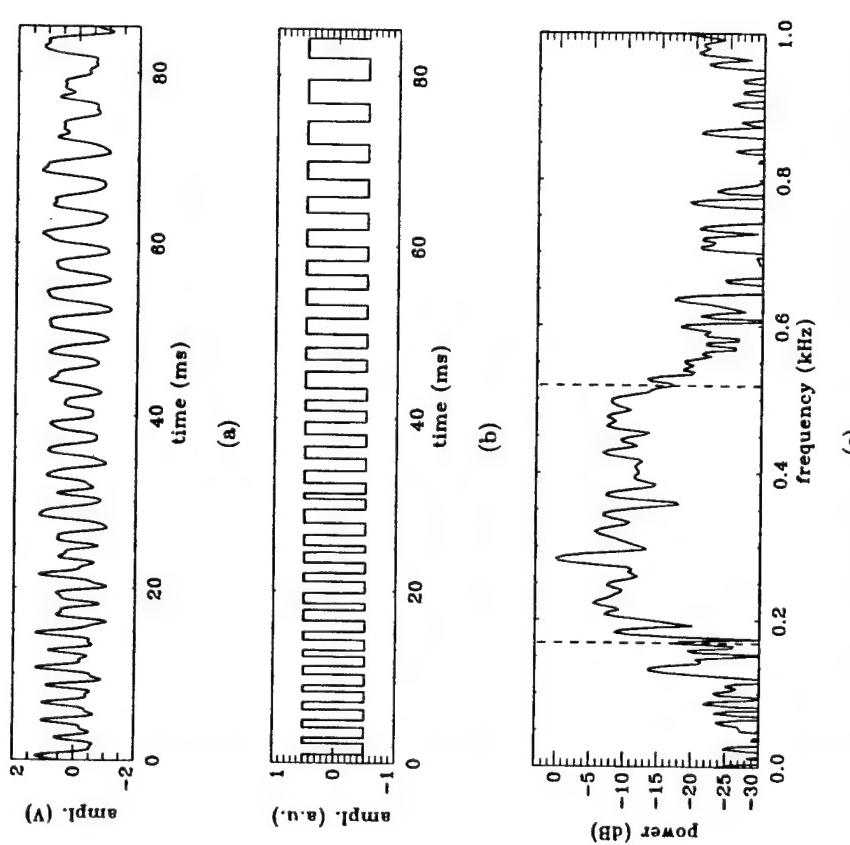
Figure 5.26: (a) Closed-loop processor output before thresholding, showing that the desired chirp is fully recovered. (b) Closed-loop processor output after thresholding. (c) Closed-loop processor output in the frequency domain, showing that a broadband signal is detected while the jammer at 281 Hz has been reduced, although it is still visible as the the dominant spectral peak. The dashed lines depict the initial and final frequencies of the original desired chirp.

Figure 5.27 shows the correlation pattern after adaptation. Note that a thin curved line is obtained this time, as should be expected from a broadband signal in the near-field (see section 5.5). Compare this figure to figure 4.51 in chapter 4 and note the remarkable similarity with the weights after adaptation in the near-field beam-forming

Figure 5.27: SLM image diffracted from the PRC, showing the correlation pattern of the near-field desired chirp.

Figure 5.28: Correlation pattern in the SLM Fourier plane, showing a systems response to a broadband signal in the near-field, and CW jammer nulling.

and CW jammer nulling simulation.

Figure 5.28 shows the correlation pattern in the Fourier plane. Compare it to figure 4.52 in chapter 4 and note that, as in the simulation, the processor responds to a circular section similar to the near-field chirp spectrum shown in figure 5.24. More remarkably, a narrow null is present in the circular section, at a temporal frequency position corresponding to that of the spatial side-lobes of the CW jammer. This null is a consequence of the adaptive response of the correlation-cancellation loop to the cross-correlation between the weights and the desired signal at that frequency. Note that in this experiment some DC light escaped through the spatial filter, recording a grating represented by a spot in the Fourier spectrum.

A question arises as to the limit of the capacity of the processor to null the CW jammer while simultaneously beam-forming towards the desired signal. In order to quantitatively answer this question, the following set of experiments was developed in which the power of the CW jammer is varied as 1, 2, 4, 8, 16 and 32 times the power of

(a) Input SINR = 0 dB

(b) Input SINR = -3.01 dB

(c) Input SINR = -6.02 dB

Figure 5.30: CW jammer nulling and near-field beam-forming: spectra of the output of the processor for input SINRs ranging from 0 dB to -6.02 dB.

the input SINR decreases, evidenced by the dimming of the broadband section in the Fourier plane. How does the decrease in the input SINR affect the performance of the processor? This questions is answered by the plots in figures 5.30 and 5.31, which show

Input SINR = 0 dB

Input SINR = -3.01 dB

Input SINR = -6.02 dB

Input SINR = -9.03 dB

Input SINR = -12.0 dB

Input SINR = -15.0 dB

Figure 5.29: The Fourier plane of the input scrolling SLM for different jammer/desired signal power ratios (assuming jammer power $\gg$ noise power).

the desired signal, producing the set of input SLM frames shown in the Fourier plane in figure 5.29. Note that the jammer becomes increasingly predominant over the signal as

(a) Input SINR = -9.03 dB

(b) Input SINR = -12.0 dB

(c) Input SINR = -15.0 dB

Figure 5.31: CW jammer nulling and near-field beam-forming: spectra of the output of the processor for input SINRs ranging from -9.03 dB to -15.0 dB.

where $SINR_i$ is the SINR at the input plane. The graph in figure 5.32 shows the fractional power due to the jammer (solid) and desired (dashed) signals as a function of input SINR. At each experiment the gain was set to the value which provided the best

the spectrum of the output of the processor after adaptation for each case. The graphs are normalized about their maximum value, so that the unwanted peak corresponding to the CW jammer is always at 0 dB. However, note that the distance between the jammer peak and the broadband chirp spectrum steadily increases as the input SINR decreases. To quantify the decrease in performance, the output signal is correlated with the desired and the jammer signal in order to determine output SINR, given by

$$SINR_o = \frac{P_d}{P_j} = \frac{\eta_d P_o}{\eta_j P_o} = \frac{\eta_d}{\eta_j},$$ (5.18)

where P_d and P_j are the output powers due to the desired and jammer signal, respectively. However, these powers are given by fractions of the constant output power $P_o = P_d + P_j$, where η_d and η_j are the fraction of the total output power due to the desired signal and to the jammer, respectively. It is assumed that the jammer power is much larger than the noise power, allowing additive noise to be ignored. The fractional power values are estimated from the output time series by using the following relations:

$$\eta_d = \frac{E[|r_{od}(\kappa)|^2]}{E[|o(\kappa)|^2] E[|d(\kappa)|^2]} - \frac{E[|r_{dj}(\kappa)|^2]}{E[|d(\kappa)|^2] E[|j(\kappa)|^2]}$$ (5.19)

$$\eta_j = \frac{E[|r_{oj}(\kappa)|^2]}{E[|o(\kappa)|^2] E[|j(\kappa)|^2]} - \frac{E[|r_{jd}(\kappa)|^2]}{E[|d(\kappa)|^2] E[|j(\kappa)|^2]}$$ (5.20)

where $E[|r_{od}(\kappa)|^2]$ is the power of the cross-correlation between the output and the desired signal, r_{oj} is the cross-correlation between the output and the jammer signal, and $r_{dj} = r_{jd}$ is the cross-correlation between the desired signal and the jammer, $E[|d(\kappa)|^2]$ is the power of the jammer signal, and $E[|j(\kappa)|^2]$ is the power of the output signal, $E[|o(\kappa)|^2]$ is the power of the output signal. The second term in each equation is added in order to account for the presence of cross-correlations between the jammer and desired signal in the output signal. The signals are assumed to be ergodic, such that the expected values can be estimated from the average of the time series. The processing gain P_g of the system is then defined by

$$P_g = \frac{SINR_o}{SINR_i},$$ (5.21)

Figure 5.32: Fractional output power due to the jammer (solid line) and desired signal (dashed line) as a function of input SINR.

Figure 5.33: Output SINR as a function of input SINR.

Figure 5.34: Processing gain as a function of input SINR, showing a nearly linear increase in jammer nulling as the jammer becomes increasingly stronger.

output SINR, such that the fractional desired signal power remained almost constant. Higher loop gain increases the power in the detected desired signal, but it also increases the power in the detected jammer. Figure 5.33 compares the performance at each experiment by plotting the output SINR as a function of the input SINR. Note that

initially the output SINR decreases slowly until the input SINR reaches a value of -6 dB, when it decreases at a rate of 4.67 dB per decade until an input SINR of -12 dB, when the decay rate once again decreases. Initially, the processor has enough degrees of freedom to compensate for the CW jammer, and is thus capable of canceling the jammer enough to maintain a high output SINR. However, as the jammer gets larger (and its corresponding eigenvalue becomes larger) the weights start to lack the necessary dynamic range to null it. Eventually, the fractional power of the jammer becomes so large that the processor becomes easily capable of nulling at least a fraction of it, since its large corresponding eigenvalue in the correlation matrix determines a fast convergence of the weights towards nulling the strong CW jammer (see section 4.2.2 in chapter 4). This behavior is more evident in figure 5.34. Note that even thought the output SINR decreases with decreasing input SINR, the processing gain becomes continually larger, reflecting the fact that it is easier to null a stronger jammer.

5.4.3 Multiple beam-forming in a circular array

In this scenario, a circular array is used in which the 256 array elements are placed in a planar circle, equally separated by an angle $\beta = 2\pi/N = 24.5$ mrad. The signal from each antenna element is sequentially fed to the processor counter-clockwise, starting from the angle $\beta_0 = 0$, such that the signal in the ith row of the SLM will contain the signal $\bar{s}_i$ detected at an angle $\beta_i = i\beta$. The processor performs broadband beam-forming towards two chirp signals both at the same frequency range as before, located in the far-field (first chirp AOA = 0.25 radians, second chirp AOA = -2 radians) (see Fig. 4.54). Figure 5.35 shows one of the frames displayed in the SLM, where sinusoidal patterns are visible due to the linear rastering of the circular array into the rectangular SLM. Figure 5.36 shows the spectrum in the Fourier plane of the SLM. Note the presence of a broad spectrum occupying a range of spatial frequencies. Antenna element functions with directional selectivity would paint an amplitude distribution across the array of

Figure 5.35: One of the 128 images captured from the SLM, showing two broadband chirps as detected by a circular array.

Figure 5.36: Image of the SLM in the Fourier plane, showing a uniform signal distribution over all possible angles of arrival, characteristic of a circular array.


Figure 5.37: (a) Processor output before thresholding, showing that the desired chirp is fully recovered. (b) Processor output after thresholding. (c) Closed-loop processor output in the frequency domain, showing that a broadband signal is detected in the same frequency range as the desired signal. The dashed lines depict the initial and final frequencies of the original desired chirp.

elements with maximum power received from the antenna pointed towards the received signal. However, we are simulating the case of vertical dipoles with no in-plane array function. As such, a far-field wavefront going towards the array will be detected by every antenna element with a time-delay proportional to the propagation time to that element. Under these circumstances even a signal in the far-field presents a broad spatial spectrum comprised of the superposition of the straight lines corresponding to the spectrum of the signal detected by each antenna element.

Figure 5.37 shows a typical waveform of the detected output before adaptation and thresholding, plotted in the time domain before thresholding (a), after thresholding (b), and in the frequency domain (c). Even though they overlap in real space and Fourier space, the desired chirps are clearly recovered and coherently added in order to provide a strong and clean output signal. The BER for this scenario is 6.20×10^{-3}.

Figure 5.38 shows the correlation pattern after adaptation. Note that two sinusoidal correlation patterns are present, each corresponding to a response towards each one of the individual chirps. The phase of each sinusoid is shifted according to the

AOA of the far-field source (see Section 5.5 for more details). Compare this figure to figure 4.57 and note the remarkable similarity with the weights after adaptation in the simulation of multiple beam-forming in a circular array.

Figure 5.39 shows the correlation pattern in the Fourier plane. Compare it to

first aliased order

spectrum of chirp and
broadband jammer

$\beta = 0$

π

π

Figure 5.41: Image of the SLM in the Fourier plane, showing a uniform signal distribution over all possible transverse spatial frequencies characteristic of a circular array.

antenna elements

time delay

Figure 5.40: One of the 128 images captured from the SLM, showing a broadband chirp and a broadband jammer detected by a circular array.

Figure 5.39: Correlation pattern in the Fourier plane of the SLM, showing a response towards broadband signals over all transverse spatial frequencies.

antenna elements

time delays

τ

Figure 5.38: Image of the SLM diffracted from the PRC, showing the correlation pattern of the two chirps detected by a circular array.

plane of the SLM. Note the presence of a broad spectrum occupying all possible spatial frequencies. For the circular array, it is no longer possible to clearly distinguish the chirp spectrum from the jammer spectrum by looking in the Fourier plane. Fortunately, this does not become a limitation to the performance of the processor, as can be seen from the plots of the output waveforms shown in figure 5.42.

Figure 5.42 shows a typical waveform of the detected output after adaptation and thresholding, plotted in the time domain before thresholding (a), after thresholding (b), and in the frequency domain (c). The desired chirp is clearly recovered. The overlapping of the temporal and spatial spectra does not limit the performance since the processing is simultaneously performed over a linear combination of all input channels, at all frequencies. Besides, the AOAs of the signal are well separated, making it easier for the array to distinguish between each signal. A problem arises, however, when the jammer is broadband and repetitive with the same periodicity as the desired signal, as in our case due to the limitations of our SLM driver. The broadband jammer and the desired signal become correlated and the processor will adapt in order to perform

figure 4.58 and note that, as in the simulation, the processor responds to a broadband signal within a range of spatial frequencies, as should be expected in a circular array.

5.4.4 Broadband beam-forming and jammer nulling in a circular array

In this scenario, a circular array is used, and as before the 256 array elements are placed in a planar circle, equally separated by an angle $\beta = 2\pi/N = 24.5$ mrad and these signals are sequentially fed into the columns of the SLM delay-lines. The processor performs broadband beam-forming on a chirp signal which is at the same frequency range as before, located in the far-field (AOA = 0.2 radians). Simultaneously, it nulls out a broadband jammer generated from Gaussian white noise, filtered to be in the same frequency range as the chirp, and located in the far-field (AOA = -0.35 radians), with twice as much power as the desired chirp. Figure 5.40 shows one of the frames displayed in the SLM, where sinusoidal patterns are visible due to the linear rastering of the circular array into the rectangular SLM. Figure 5.41 shows the spectrum in the Fourier

antenna elements

time delays

Figure 5.43: SLM image diffracted from the PRC, showing the correlation pattern of a broadband chirp detected by a circular array.

Figure 5.44: Correlation pattern in the SLM Fourier plane, showing a response towards broadband signals over all transverse spatial frequencies.

repetitively displaying the 128 frames stored in its memory. In order to eliminate such an artifact, a better SLM driver would have to be used, capable of simulating the continuous generation of a truly broadband jammer, uncorrelated with the desired signal. Such an effect, however, does not take place in the CW jammer scenario, since no linear transformation is capable of extracting a broadband signal out of a CW signal (except at the limited overlap between their frequencies).

Figure 5.43 shows the corresponding weights after adaptation. Note the presence of a sinusoidal pattern, corresponding to a response towards the desired chirps. The peak of the pattern is at the antenna element corresponding to the AOA of the chirp.

Figure 5.44 shows the correlation pattern in the Fourier plane. Note that, as in the previous scenario, the processor responds to a broadband signal coming from all transverse spatial frequencies.

Figure 5.42: (a) Processor output before thresholding, showing that the desired chirp is fully recovered. (b) Processor output after thresholding. (c) Closed-loop processor output in the frequency domain, showing that a strong response over the frequency range of the desired signal (dashed lines). However, the jammer overlaps the same frequency range.

a linear transformation which will make the jammer resemble the desired signal as close as possible, artificially resulting in a better SINR than expected. Such an effect was simulated in section 4.3.5 in chapter 4, with similar results. The BER achieved in this scenario is 6.02×10^{-3}. It is due to a limitation in the SLM, only capable of

(a)

(b)

(c)

Figure 5.47: (a) Processor output before thresholding, showing that that the broadband chirp is mostly recovered. (b) Processor output after thresholding. (c) Output spectrum, showing a strong response in the frequency range of the desired chirp (dashed lines). However, a broadband jammer overlaps with the temporal frequency of the desired signal.

jammer). The back-reflection from the beam-splitter can also be seen, overlapping with the left-half of the signal spectrum.

Figure 5.47(a) and (b) shows the processor output after adaptation, showing that the desired signal has been recovered. As explained in section 5.4.4, since the broadband jammer is repetitive with the same period as the desired signal, it correlates

Figure 5.45: One of the 128 images captured from the SLM, containing a chirp in the near-field, a CW jammer and a broadband jammer in the far-field.

Figure 5.46: Image of the SLM in the Fourier plane, showing a circular section (near-field chirp), a strong spot (CW jammer), and a thin and long line (broadband jammer).

5.4.5   Near-field beam-forming, CW and broadband jammer nulling

In this scenario, the processor performs beam-forming towards a broadband chirp in the near-field of the antenna array (azimuthal distance = 0.5 L, range = 3.5 L) while simultaneously nulling a CW jammer in the far-field (AOA = 1.25 radians) at 281 Hz, with the same power as the desired signal. A broadband jammer is also present in the far-field (AOA = -1 radians) with twice as much power as the desired signal and in the same frequency range as the desired signal (from 168.8 Hz to 518.8 Hz). Figure 5.45 shows one of the frames displayed on the SLM. The jammers blur most of the picture, but some of the low frequency ripples of the near-field chirp can be seen. They are curved, as should be expected form a near-field signal . Figure 5.46 shows a time-averaged image captured at the Fourier plane of the SLM, showing a circular section, characteristic of a broadband signal in the near-field (the desired chirp). It also shows a strong spot, characteristic of a CW signal (the CW jammer). Finally, it shows a thin and long line, characteristic of a broadband signal in the far-field (the broadband

Figure 5.51: Image of the SLM in the Fourier plane, showing three long and thin lines (chirp, and broadband jammers) and a strong spot (CW jammer).

antenna elements

time delay

Figure 5.50: One of the 128 images captured from the SLM, showing a chirp, a CW jammer and two broadband jammers in the far-field.

antenna elements

time delays

Figure 5.48: SLM image diffracted from the PRC, showing the correlation pattern of the near-field chirp.

Figure 5.49: Correlation pattern in the SLM Fourier plane, showing the system response to the near-field chirp, broadband and CW jammers.

with the desired signal and, therefore, the processor adaptively modifies the weights in order to linearly transform the jammer into the desired signal instead of nulling it out. Consequently, instead of jamming the desired signal, the broadband far-field signal has the effect of removing some of the total signal power (which is a constant) from the desired signal, requiring the use of extra degrees of freedom in order to transform its spectrum into that of the desired signal. The CW jammer, on the other hand, cannot be transformed into the broadband chirp and must be nulled. Figure 5.47(c) shows the spectrum of the output signal, showing a strong response at the frequency range of the signal and the jammer. A peak corresponding to the CW jammer can still be seen at 281 Hz. The BER for this scenario is 6.07×10^{-3}.

Figure 5.48 shows the correlation pattern after adaptation. Note that again a thin, curved line is obtained, as expected from a broadband signal in the near-field. However, the shape of the line is different from that presented in figure 5.27, since the source is present at a different range and AOA this time. Figure 5.49 shows the

correlation pattern in the Fourier plane. Again, the processor responds to a circular section similar to the near-field chirp spectrum shown in figure 5.46. However, a null in the spectrum of the desired signal due to the CW jammer has not been detected. This can be due to the loss in degrees of freedom due to the adaptation to the near-field signal and simultaneous linear transformation of the broadband jammer. Or, more likely, due to noise and experimental error. The image may have been captured at a moment when the processor had its performance hindered by random mechanical vibrations. The PR crystal dynamically modifies its volumetric grating in order to dynamically compensate for variations in the interferometric grating. However, one can still be "unlucky" and capture the diffracted image in a moment the crystal is not fully adapted to a given variation.

5.4.6 Far-field beam-forming and multiple jammer nulling

In the final scenario, the processor performs beam-forming in the direction of a broadband chirp in the far-field of the antenna array (AOA = 0.25 radians) while

simultaneously nulling a CW jammer (f = 350 Hz) in the far-field (AOA = -0.1 radians) with the same power as the desired signal, and nulling two broadband jammers, each with twice as much power as the desired signal, and at different frequency ranges. The first broadband jammer has an AOA of 1 radian and occupies the frequency range from 168.8 Hz to 518.8 Hz (the same as the desired signal). The second broadband jammer has an AOA of -1.2 radians and occupies the frequency range from 281.3 Hz to 750 Hz (Nyquist). Figure 5.50 shows one of the frames displayed on the SLM. The jammers obscure most of the picture, but some of the low frequency ripples of the near-field chirp can be seen. They are straight, as should be expected from far-field signals. Figure 5.51 shows an image captured at the Fourier plane of the SLM, showing three long and thin lines corresponding the the far-field chirp (center one) and broadband jammers. Note that the second broadband jammer has its initial frequency at a frequency higher than the initial frequency of the desired signal, and it has a higher bandwidth. As usual, the CW jammer is represented by a strong spot in the Fourier plane. The back-reflection from the beam-splitter (left) and the first diffracted order (top) can also be seen.

Figure 5.52(a) and (b) shows the processor output after adaptation, showing that the desired signal has been recovered with a BER of 6.07×10^{-3}. Note that the second broadband jammer can only be partially transformed into the desired signal. Since its high frequency spectrum is out of the range of the desired signal, it has to be adaptively nulled. Figure 5.52(c) shows the spectrum of the output signal, showing a strong response over the frequency range of the chirp (dashed lines), except for a deep null at 350 Hz, the frequency of the CW jammer.

Figure 5.53 shows the correlation pattern after adaptation. Note that again a thin straight line is obtained, as expected from a broadband signal in the far-field. Compare this pattern to figure 4.71 in chapter 4 and note the similarity to the final weights after adaptation in the simulation of far-field beam-forming and multiple jammer nulling scenario.
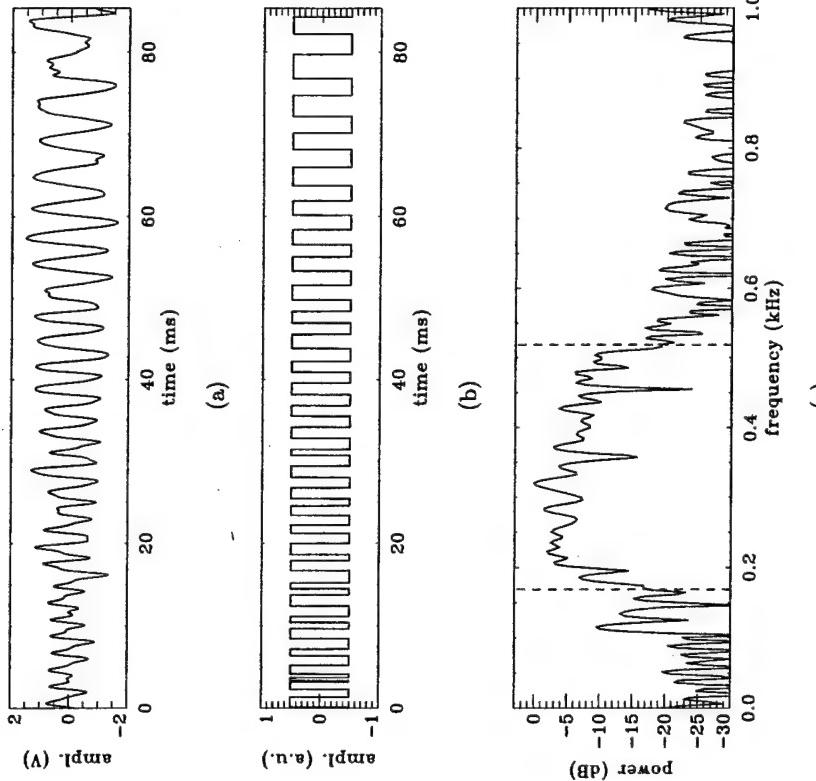
Figure 5.52: (a) Processor output before thresholding, showing that the broadband chirp is fully recovered. (b) Processor output after thresholding. (c) Output spectrum, showing the detection of a broadband signal in the frequency range of the desired chirp (dashed lines), except for a deep null at 350 Hz.

Figure 5.54 shows the correlation pattern in the Fourier plane. The processor responds strongly to the chirp and to the first broadband jammer, which is in the same frequency range as the desired signal. This is expected, since both signals correlate. The second broadband jammer, however, does not correlate well with the desired signal. Consequently, the processor is not capable of efficiently transforming it into the desired signal.

Figure 5.56: Estimation of the AOA of a signal in the far-field using its power spectrum.

Figure 5.55: Estimation of the AOA of a signal in the far-field using its correlation pattern.

Figure 5.54: Correlation pattern in the SLM Fourier plane, showing the systems response to the near-field chirp, broadband and CW jammers.

Figure 5.53: Diffracted image from the PRC in the SLM image plane, showing the correlation pattern and weights of the near-field chirp.

approximation, this pattern resembles the auto-correlation of the desired signal. It was also noted that this pattern is modified according to the AOA and range of the signal being detected. Thus, one can conclude that it is possible to use these patterns in order to determine the position of the signal source towards which the adaptive antenna array is beam-forming [115]. The spatial position of the correlation peak is determined by the relative time delay between the input signal wavefront and the desired signal, which varies according to the range from the source to a specific antenna element. For a far-field signal in an equispaced linear array, the source is effectively seen at infinity, and the desired signal auto-correlation produces a straight correlation slice tilted at an angle proportional to the sine of the AOA of the input signal, as shown in figure 5.55. That is, the relative time delay from the first antenna element to the ith antenna element is given by

$$\Delta t_i = a_i \frac{\sin \theta}{v}, \qquad (5.22)$$

where a is the inter-element antenna spacing, v is the velocity of the acoustic wave in the medium and θ is the AOA of the signal. Thus, the position of the correlation peak

signal and most of its power spectrum is lost. Compare this pattern to figure 4.72 and note the remarkable similarity. Also note the thin null at the frequency of the CW jammer due to the overlap of the spatial side-lobes of the jammer with the desired signal spectrum. As in the simulation, a spot is also present at the spectral position of the CW jammer. This is to be expected, since better nulling is achieved when the weights respond to the CW jammer with a 180° phase inversion. This remarkable result also explains the deep null present in the output spectrum at the jammer frequency. Unfortunately, such a result is only sporadically achieved, since long term stability in the PR holographic recording is required for the continuous accumulation of the coherent CW grating necessary for producing such deep nulls.

5.5   Source position estimation

As shown in the previous scenarios, it is possible to obtain a pattern proportional to the correlation between the desired signal and the error signal. To a first order

is given by the relation

$$\tau = \frac{t_0}{\Delta_t} + \left(\frac{a\sin\theta}{v\Delta_t}\right)i, \qquad (5.23)$$

where t_0/Δ_t is a constant time delay (the intercept of the vertical axis of Fig. 5.55), given by the relative time delay t_0 between the desired signal present at the first antenna element ($i = 0$) and the beginning of the desired signal present at the feedback error signal. $\frac{a\sin\theta}{v\Delta_t}$ is the slope of the straight line representing the correlation slice. In the auto-correlation of a broadband signal, the result is a slice with a width proportional to one over the bandwidth of the original signal, with sinc-like side-lobes, just as shown in figure 5.55. For a correlation pattern with slope $\tan\alpha$, the AOA of the corresponding signal source is given by

$$\theta = \arcsin\left(-\frac{\tan(\alpha)v\Delta_t}{a}\right). \qquad (5.24)$$

Note that the term $v\Delta_t/a$ is a constant parameter for a given array. Thus, the slope of the correlation slice is only a function of the AOA of the signal source. For the case depicted in figure 5.55, the AOA is $\pi/4$. Note that we are assuming square pixels, which is the case for our SLM. Otherwise, the expression inside the brackets would have to be multiplied by the aspect ratio of the SLM.

Figure 5.56 shows that the AOA can also be determined from the tilt of the power spectrum of the signal. Since the power spectrum is given by the Fourier transform of the correlation pattern, we have that α_1 will be complementary to α ($\alpha_1 = 90° - \alpha$). Thus, the AOA is given by

$$\theta = \arccos\left(-\frac{\tan(\alpha_1)v\Delta_t}{a}\right). \qquad (5.25)$$

In the near-field case, the range from the source to each antenna element varies quadratically and the resulting correlation slice is curved, as seen in figure 5.57. In this case, the relative propagation time delay from the center antenna element to another
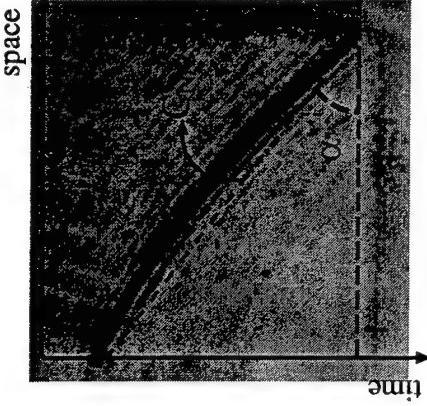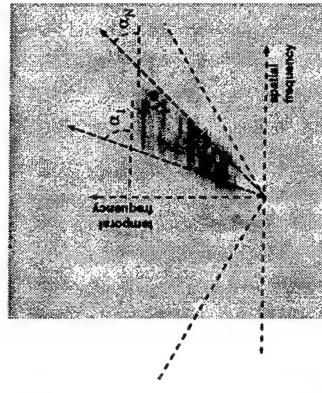
Figure 5.57: Estimation of the AOA of a signal in the near-field using its correlation pattern.

Figure 5.58: Estimation of the AOA of a signal in the near-field using its power spectrum.

antenna element i in an equidistant, linear array, is given by

$$t_i = \frac{\sqrt{x^2 + \left[y - \left(\frac{N}{2} - i\right)a\right]^2}}{v}, \qquad (5.26)$$

where x represents the range and y represents the azimuthal distance from the signal source to the center of the antenna array. The spatial position of the correlation peak is given by the relation

$$\tau = \frac{t_0}{\Delta_t} + \frac{\sqrt{x^2 + \left[y - \left(\frac{N}{2} - i\right)a\right]^2}}{v\Delta_t}. \qquad (5.27)$$

Expressing the source position in polar coordinates, we have

$$t_i = \frac{\sqrt{r^2 - 2iar\cos\theta - i^2a^2}}{v}, \qquad (5.28)$$

where the source position is given by (r, θ), with respect to the leftmost antenna element. The exact position of the source is then given by:

$$\tan\alpha = \frac{1}{\Delta_t vN}\left(\sqrt{r^2 - 2Lr\cos\theta + L^2} - r\right), \qquad (5.29)$$

Figure 5.59: Estimation of the AOA of a signal in a circular array using its correlation pattern.

In the circular case, with the signal source in the far-field, the resulting correlation slice is sinusoidal, as seen in figure 5.59. If the N antenna elements are evenly spaced about a circle of radius R, the angle between the central antenna element (at 0 degrees) and antenna element i is given by

$$\beta_i = \frac{2\pi i}{N} \qquad (5.34)$$

and the propagation time delay from the center of the array to element i is given by

$$t_i = \frac{R}{v}[1 - \cos(\beta_i - \theta)], \qquad (5.35)$$

and the spatial position of the correlation peak is given by the relation

$$\tau = \left(\frac{t_0}{\Delta_t} + \frac{R}{v\Delta_t}\right) - \frac{R}{v\Delta_t}\cos(\beta_i - \theta), \qquad (5.36)$$

$$C_{N/2} = \frac{r}{\Delta_t v}\left(\frac{1}{2} + \frac{1}{2}\sqrt{1 - \frac{2L}{r}\cos\theta + \frac{L^2}{r^2}} - \sqrt{1 - \frac{L}{r}\cos\theta + \left(\frac{L}{2r}\right)^2}\right), \qquad (5.30)$$

where α is the angle between the antenna array and line segment joining the two extrema of the correlation slice, $L = Na$ is the total length of the array, and $C_{N/2}$ is the sag between the correlation slice and the straight line segment, at the center of the array ($i = N/2$). Note that whenever an inflection point is present at the near-field correlation slice, the signal source is located directly ahead of the corresponding antenna element.

Using the binomial approximation for the square root, the following solution is obtained, valid for $r > L$:

$$r = \frac{L^2}{8v\,\Delta_t\,C_{N/2}}, \qquad (5.31)$$

$$\theta = \arccos\left[\frac{rL}{2} - \frac{v\,\Delta_t\tan\alpha}{a}\right]. \qquad (5.32)$$

In the case shown in figure 5.57, we have that $\Delta_t = 0.667$ msec/sample, $a = 51.85$ mm, $L = Na = 13.2$m, $v = 0.3$ m/msec, $\alpha = 37°$ and $C_{N/2} = 12$ samples (actually 24 samples, but the 256 pixels in the temporal dimension are used to represent 128 time-delays), giving us $r = 4.54L$ and $\theta = 1.27$ radians, when the actual values used to generate the input data are $r = 4.35L$ and $\theta = 1.17$ radians, showing reasonable position estimation accuracy.

Figure 5.58 shows that the source position can also be estimated from the power spectrum of the diffracted weights. In this case, θ_1 (the AOA with respect to the leftmost antenna element, $i = 1$) is determined by equation 5.25 using the slope at the left side of the circular section. Similarly, θ_N (the AOA with respect to the rightmost antenna element, $i = N$) is also determined by equation 5.25, this time using the slope at the right side of the circular section. Then, the range r is given by triangulation from the leftmost and rightmost AOAs:

$$r = \frac{L\tan\theta_N}{\cos\theta_1\tan\theta_N - \sin\theta_1} \qquad (5.33)$$

corresponding to a cosinusoid translated by $\frac{f_0}{\Delta_t} + \frac{R}{v\Delta_t}$ and with a peak in the direction of the antenna element i with an angle β_i closest to the AOA θ of the incoming wave.

In the example shown in figure 5.59 (the same shown in section 5.4.3), multiple beam-forming is performed in the direction of two desired beams located in the far-field of the antenna array. Reading the correlation slices we find that the first signal has an AOA $\cong \pi/8 = 0.39$ radians, and the second signal has an AOA $\cong -\pi/2 - \pi/8 = -1.96$ radians, when the actual values used to generate the input data are 0.25 radians for the first signals and -2 radians for the second signal.

5.6 Summary

This chapter presented and analyzed an experimental implementation of an input delay plane FIRNN architecture used as an adaptive antenna array processor. The resulting processor proved to be capable of beam-forming and jammer nulling in the presence of multiple broadband signals and jammers, in the far-field and near-field. Such adaptability is possible thanks to the spatio-temporal signal processing capability provided by FIRNNs with their associated degrees of freedom in the spatial domain and temporal domain. Experimental results were presented for equispaced linear and circular arrays. However, conformal arrays could also have been used with similar results.

Some important conclusions could be drawn from the experimental implementation. They are:

(1) Commercially available high speed FLC-SLMs can be used to simulate multi-channel delay-lines.

(2) A method for active stabilization is imperative for long-term interferometric detection.

(3) Time-multiplexing can be successfully employed for holographic read/write iso-lation

(4) RF mixing can be used to provide a lock-in reference to signals optically mixed by AO modulators.

The SLM driver proved to impose several limitations to the performance of the processor. These limitations are:

(1) The highest frequency of the signals being processed is limited by the speed of the SLM driver (2.5 kHz, one half the FLC-SLM switching frequency).

(2) The total number of time-delay taps is limited by the number of frames in the memory of the driver (128). Otherwise, up to 256 taps could be present.

(3) Because of the limited memory of the SLM driver, repetitive signals must be used. This limitation makes it impossible to simulate jammers which are truly uncorrelated with the desired signal. However, this is not a limitation of the processor itself, which would be capable of effectively nulling real time non-repetitive jammers.

Other minor limitations to the performance of the processor are the binary quantization of the SLM pixels, the interferometric sensitivity to vibrations of the holographic recording, and the limited dynamic range of the weights in the PRC due to the photorefractive decay. Those limitations are reflected by a reduction in the final SINR of the processor.

Finally, even though the sonar processor was devised with the communication scenario in mind, it was shown that the correlation pattern and power spectrum images obtained from photorefractive diffraction can be used to successfully estimate the position of the signal source, extending its application to other areas, such as sonar.

Chapter 6

Conclusions and future scope

6.1 Summary

This thesis presented the results of research focused on the use of finite impulse response neural networks for optoelectronic signal processing, starting from the FIRNN algorithmic description, followed by the analyses and simulations of different optoelectronic architectures, evolving to a final experimental implementation. Two gradient descent based algorithms were studied: Wan's temporal back-propagation [2] and the novel delayed feedback back-propagation algorithm. Based on these algorithms, the input delay plane and the output delay plane optical architectures were developed and it was shown how they could be efficiently combined in order to implement forward and back-propagation in multi-layer FIRNNs by simultaneously addressing the weights represented by a grating stored in the same photorefractive crystal.

However, multi-layer FIRNNs still proved to be quite complex to implement in real-time analog optoelectronic hardware. Thus, emphasis was given to the optical implementation of single-layer FIRNNs. Two single-layer architectures were simulated, one based on the delayed feedback algorithm, known as the Broadband and Efficient Adaptive Method for True-time-delay Array Processing (BEAMTAP) optoelectronic architecture, and the other based on the conventional temporal back-propagation algorithm, known as the Finite Impulse Response Optical Neural Network (FIRONN) architecture. The simulations focused on the processing of signals from adaptive an-

tenna arrays — an application which takes advantage of the spatio-temporal processing capabilities of single-layer FIRNNs. A graphic user interface (GUI) oriented software package was developed in IDL, allowing for the flexible simulation of different scenarios, such as multiple jammers with arbitrary input power, angle of arrival, bandwidth, etc., providing us also with the input data used in the experimental setup. The simulation package also includes simple models for some of the relevant physical parameters, such as input non-linearities and photorefractive weight decay. The resulting simulations were used to study the effect of different parameters over adaptation, such as the relation between the number of jammers and the final signal to interference plus noise ratio after adaptation. The effect of weight decay was also studied, resulting in a mathematical expression used to determine the final weights as a function of the optimum weights, the learning rate, the decay rate and the eigenvalues of the auto-correlation function, showing that larger eigenvalues are the least affected by weight decay.

Since an optoelectronic implementation of the BEAMTAP architecture is already underway [74], emphasis was given to an experimental implementation and demonstration of the FIRONN architecture. Experimental results were presented for an optical processor making use of a scrolling spatial light modulator (SLM) for representing the input signal and its delayed versions, photorefractive dynamic gratings for implementing the adaptive weights, differential heterodyning for bipolar signal representation, a phase-locked loop for controlling the optical path length, providing long term interferometric stabilization, and acousto-optic modulators for modulating the feedback error signal. Results for a combination of multiple beam-forming and jammer nulling were presented for planar and circular adaptive arrays, with signals present in the far-field and in the near-field. Succesful beam-forming was observed in all scenarios. Jammer nulling, however, was only effectively observed in the case of narrowband jammers. The limited memory of the scrolling SLM driver made it necessary to repeat the input frames after 128 samples, resulting in a partial correlation between the desired signal

and broadband jammers, allowing the processor to linearly transform the jammers into the desired signal rather than nulling them.

An experiment was performed in order to determine the relationship between the intensity of a CW jammer and the processing gain of the processor, showing that the processing gain increases roughly linearly with the intensity of the jammer. Finally, it was also shown how one could estimate the position of the signal source from the SLM images diffracted from the photorefractive hologram used to store the dynamic weights. The estimation methods proved to be effective in the estimation of the position of signal sources in the far-field and near-field of a planar array, and in the far-field of a circular array.

The FIRONN optoelectronic processor was successfully analyzed, implemented, and its functionality was verified by the experimental results obtained. The experimental results proved to be remarkably similar to the simulations, verifying that most of the relevant physical parameters had been included in the simulation model. Some of the main goals attained by the experimental implementation were: 1 — a demonstration of the effectiveness of photorefractive grating formation for weight adaptation, and 2 — the creation of a test-bed capable of easily processing sonar array data under different scenarios.

6.2 Future directions

I expect that the work presented on this thesis will be extended and advanced. Thus, I propose the following suggestions for future extensions. Although improvements could be made to the FIRNN algorithms, optical architectures and simulations, it is certainly the experimental implementation that inspires most attention, since it represents the final goal of the present study, and also validates all the analyses and simulations that preceded it. The most relevant suggestions pertaining to the experimental implementation are listed next:

(1) **Improved SLM driver:** A simple way to improve the performance of the experimental demonstration would be to replace the current SLM driver with an improved model with twice as much memory, allowing the use of 256 taps instead of 128. Better yet, a driver that could accept data in real-time could be used. This would overcome the problem arising from the partial correlation between the desired signal and repeating jammers. Such a driver could also be used for probing the processor output in a more elaborate form, thus producing an output proportional to the angular and frequency receptivity pattern of the array, for example.

(2) **Custom design of a scrolling SLM:** Instead of loading a completely new frame at every time-step, an SLM could be custom designed by using a shift register at each of its columns, thus making it a truly scrolling SLM. This way, it would only be necessary to load a single row at every time-step and the subsequent rows would be derived by delayed versions of the first row. This modification would allow a faster operation of the SLM (up to 5 kHz), limited only by the switching time of the bistable FLC, instead of being limited by the time necessary to load the SLM with 65536 new bits at every time-step. This modification would also simplify the design of the SLM driver, since much less memory would be required for storing the same number of frames, and a single row would need to be loaded every time. Finally, the less stringent requirement on the driver operation would make it easier to address SLMs with higher resolution, increasing the number of degrees of freedom of the processor and the number of operations performed optically in parallel at every time-step.

(3) **Increased operating frequency:** at the moment the operating frequency of the processor is limited by the switching speed of the scrolling SLM. The previous suggestion would already improve its performance by a factor of three

or more. However, alternative SLM technologies such as the quantum well modulators mentioned in chapters 3, 4 and 5 could be used to improve the performance by at least three orders of magnitude, perhaps enough for achieving bandwidths in the GHz range. Alternative technologies could also be used to overcome the limitation imposed by the bistable operation of ferroelectric SLMs, which forces the binary quantization of the input data.

A more ambitious but worthwhile goal would be to implement one of the multi-layer architectures presented in this thesis. Such a goal would almost certainly require the miniaturization of the optomechanical components, perhaps by using slotted base-plates in order to maintain some flexibility of implementation.

Bibliography

[1] P. E. X. Silveira and K. H. Wagner, "Time delay optical neural network," Proc. Int. Conf. on Optics in Computing, vol. 34, no. 90, pp. 266–269, 1998.

[2] E. A. Wan, "Temporal backpropagation for FIR neural networks," in Proc. Int. Joint Conf. Neural Networks, June 1990, pp. I 575–580.

[3] P. Werbos, Beyond Regression: New tools for prediction and analysis in the behavioral sciences. PhD thesis, Harvard University Committee on Applied Mathematics, November 1974.

[4] P. E. X. Silveira and K. H. Wagner, "Optical architecture for finite impulse response neural networks," in Proc. Int. Conf. on Optics in Computing, SPIE, 1999, pp. 190–192.

[5] L. J. G. B. Widrow, P. E. Mantey and B. B. Goode, "Adaptive antenna systems," Proceedings of the IEEE, vol. 55, no. 12, pp. 2143–2161, 1967.

[6] K. H. Wagner, S. Kraut, L. Griffiths, S. Weaver, R. Weverka, and A. Sarto, "Efficient true-time-delay adaptive array processing," in Proc. SPIE 2845, IEEE, 1996.

[7] A. Yariv, Optical Electronics in Modern Communications. Oxford University Press, fifth ed., 1997.

[8] J. Rhodes, "Adaptive filter with a time-domain implementation using correlation cancellation loops," Applied Optics, vol. 22, no. 2, pp. 282–287, 1983.

[9] R. M. Montgomery and M. R. Lange, "Photorefractive adaptive filter structure with 40-dB interference rejection," Applied Optics, vol. 30, no. 20, pp. 2844–2849, 1991.

[10] H. Toyoda, N. Mukohzaka, Y. Suzuki, and M. Ishikawa, "Adaptive optical-processing system with optical associative memory," Applied Optics, vol. 32, no. 8, pp. 1354–1358, 1993.

[11] D. Psaltis and Y. Qiao, "Adaptive multilayer optical networks," Progress In Optics, vol. 31, pp. 227–261, 1993.

[12] Z. Gan and D. Z. Anderson, "Acoustic signal recognition with a photorefractive time-delay neural network," Optics-Lett, vol. 19, no. 9, pp. 655–657, 1994.

[13] D. A. Pierre, Optimization theory with applications. John Wiley & Sons, 1969.

[14] D. G. Luenberger, Introduction to linear and nonlinear programming. Addison Wesley, 1973.

[15] R. Fletcher, Practical methods of optimization. John Wiley & Sons, 1990.

[16] C. Bishop, Neural Networks for Pattern Recognition. Clarendon Press, 1995.

[17] A. Cauchy, "Méthode générale pour la résolution des systémes d'équations simultanées," Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences, vol. 25, pp. 536–538, 1847.

[18] B. Widrow and S. D. Stearns, Adaptive Signal Processing. Prentice Hall, 1985.

[19] P. S. R. Diniz, Adaptive Filtering: Algorithms and Practical Implementation, ch. 1, pp. 8–13. Kluwer Academic Publishers, 1997.

[20] S. U. H. Qureshi, "Adaptive equalization," Proc. IEEE, vol. 73, no. 9, pp. 1349–1387, 1985.

[21] B. Widrow and E. Walach, Adaptive inverse control. Prentice-Hall, 1996.

[22] J. Makhoul, "Linear prediction: a tutorial review," Proc. IEEE, vol. 63, no. 4, pp. 561–580, 1975.

[23] J. G. Proakis and M. Salehi, Communication Systems Engineering. Prentice-Hall, 1994.

[24] R. Duda and P. Hart, Pattern Classification and Scene Analysis. New York: Wiley, 1973.

[25] Y. Le Cun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural Computation, vol. 1, pp. 541–551, 1989.

[26] E. Hartman, J. Keeler, and J. Kowalski, "Layered neural networks with gaussian hidden units as universal approximations," Neural Computation, vol. 2, pp. 210–215, 1990.

[27] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," Neural Networks, vol. 2, pp. 359–366, 1989.

[28] I. W. T. Miller, R. S. Sutton, and P. J. Werbos, eds., Neural Networks for Control. The MIT Press, 1990.

[29] T. Kohonen, "An adaptive associative memory principle," IEEE Transactions on Computers, vol. C-23, pp. 444–445, 1974.

[30] T. Kohonen, Self-Organization and Associative Memory. Berlin: Springer-Verlag, 3 ed., 1989.

[31] D. E. Rumelhart, R. Durbin, R. Golden, and Y. Chauvin, Backpropagation: theory, architectures, and applications, pp. 1–34. Lawrence Erlbaum, 1995.

[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representation by error propagation," in D. E. Rumelhart and J. L. McClelland, eds., (Parallel Distributed Processing: Explorations in the Microstructure of Cognition), ch. 8. Cambridge, MA: MIT Press, 1986.

[33] A. H. Waibel, T. Hanazawa, G. E. H. K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 37, no. 3, pp. 328–339, 1989.

[34] K. J. Lang, A. H. Waibel, and G. E. Hinton, "A time-delay neural network architecture for isolated word recognition," Neural Networks, vol. 3, pp. 23–43, 1990.

[35] E. A. Wan, "Time series prediction by using a connectionist network with internal delay lines," in A. S. Weigend and N. A. Gershenfeld, eds., (Time Series Prediction: Forecasting the Future and Understanding the Past), SFI studies in the science of complexity. Addison-Wesley, 1993.

[36] H. Gleisner, H. Lundstedt, and P. Wintoft, "Predicting geomagnetic storms from solar-wind data using time-delay neural networks," Annales Geophysicae, vol. 14, pp. 676–686, 1996.

[37] D. S. Clouse, C. L. Giles, B. G. Horne, and G. W. Cottrell, "Time-delay neural networks - representation and induction of finite-state machines," IEEE Transactions On Neural Networks, vol. 8, no. 5, pp. 1065–1070, 1997.

[38] Y. Tan and A. V. Cauwenberghe, "Nonlinear one-step-ahead control using neural networks: Control strategy and stability design," Automatica, vol. 32, no. 12, pp. 1701–1706, 1996.

[39] F. Lavagetto, "Time-delay neural networks for estimating lip movements from speech analysis - a useful tool in audio-video synchronization," IEEE Transactions On Circuits And Systems For Video Technology, vol. 7, pp. 786–800, 1997.

[40] P. R. Chang, C. G. Lin, and B. F. Yeh, "Inverse filtering of a loudspeaker and room acoustics using time-delay neural networks," Journal Of The Acoustical Society Of America, vol. 95, pp. 3400–3408, 1994.

[41] P. E. X. Silveira, G. S. Pati, and K. H. Wagner, "Optical implementation of a single-layer finite impulse response neural network," in Proc. Int. Conf. on Optics in Computing, Vol. 4089, SPIE, June 2000, pp. 656–667.

[42] W. F. Gabriel, "Adaptive arrays - an introduction," Proc. IEEE, pp. 239–272, 1976.

[43] R. A. Sprague and C. L. Koliopoulos, "Time integrating acoustooptic correlator," Applied Optics, vol. 15, no. 1, pp. 89–92, 1976.

[44] P. Kellman, "Time integrating optical signal processing," Optical Eng., vol. 19, no. 3, pp. 370–375, 1980.

[45] N. A. Riza, "Space integrating interferometric acousto-optic convolver," IEEE Photonics Technology Letters, vol. 7, no. 3, pp. 339–341, 1995.

[64] L. Arizmendi, P. D. Townsend, M. Carrascosa, J. Baquedano, and J. M. Cabrera, "Photorefractive fixing and related thermal effects in LiNbO3," J-Phys-Cond, vol. 3, pp. 5399–5406, 1991.

[65] D. Dolfi, F. Michelgabriel, S. Bann, and J. P. Huignard, "2-dimensional optical architecture for time-delay beam forming in a phased-array antenna," Optics Letters, vol. 16, pp. 255–257, 1991.

[66] N. A. Riza, "Transmit-receive time-delay beam-forming optical architecture for phased-array antennas," Applied Optics, vol. 30, pp. 4594–4595, 1991.

[67] X. S. Yao and L. Maleki, "A novel 2-d programmable photonic time-delay device for millimeter-wave signal-processing applications," IEEE Photonics Technology Letters, vol. 6, pp. 1463–1465, 1994.

[68] A. P. Goutzoulis and D. K. Davies, "Hardware-compressive 2-d fiber optic delay line architecture for time steering of phased-array antennas," Applied Optics, vol. 29, no. 36, pp. 5453–5359, 1990.

[69] E. Toughlian and H. Zmuda, "Variable time-delay system for broadband phased array and other transversal filtering applications," Optical Engineering, vol. 32, no. 3, pp. 613–617, 1993.

[70] M. Y. Frankel and R. D. Esman, "Dynamic null steering in an ultrawideband time-steered array antenna," Applied Optics, vol. 37, no. 23, pp. 5488–5494, 1998.

[71] R. M. Montgomery, "Acousto-optic/photorefractive processor for adaptive antenna arrays," Optoelectronic Signal Processing for Phased-Array Antennas II, pp. 207–217, 1990.

[72] C. W. Keefer, J. E. Malowicki, and P. M. Payson, "Wideband operation of a photorefractive based adaptive processor," in Proc. of SPIE, Vol. 1790, 1992, pp. 145–156.

[73] A. P. Goutzoulis, Transducer Design, pp. 285–337. Marcel Dekker, Inc., 1994.

[74] G. Kriehn, A. Kiruluta, P. E. X. Silveira, S. Weaver, S. Kraut, K. Wagner, R. T. Weverka, and L. Griffiths, "Optical beamtap beam-forming and jammer-nulling system for broadband phased-array antennas," Applied Optics, vol. 39, pp. 212–230, January 2000.

[75] P. E. X. Silveira, G. S. Pati, and K. H. Wagner, "Optoelectronic FIR neural networks," to be published in Applied Optics.

[76] D. R. Pape, "Multichannel bragg cells - design, performance, and applications," Optical Engineering, vol. 31, pp. 2148–2158, 1992.

[77] D. A. B. Miller, "Quantum-well self-electro-optic effect devices," Optical and Quantum Electronics, vol. 22, pp. S61–S98, 1990.

[46] A. Yariv and P. Yeh, Optical waves in crystals. John Wiley & Sons, 1984.

[47] P. Gunter and J. P. Huignard, eds., Photorefractive materials and their applications I. Springer Verlag, 1988.

[48] P. Gunter and J. P. Huignard, eds., Photorefractive materials and their applications II. Springer Verlag, 1989.

[49] P. Yeh, Introduction to photorefractive nonlinear optics. John Wiley & Sons, 1993.

[50] L. Solymar, D. J. Webb, and A. Grunnet-Jepsen, The physics and applications of photorefractive materials. Clarendon Press, 1996.

[51] H. Kogelnik, "Coupled wave theory for thick hologram gratings," Bell Syst. Tech. J, vol. 48, pp. 2909–2947, 1969.

[52] D. Psaltis, J. Yu, and J. Hong, "Bias-free time-integrating optical correlator using a photorefractive crystal," Appl-Optics, vol. 24, pp. 3860–3865, 1985.

[53] C. X.-G. Gu, Optical neural networks using volume holograms. PhD thesis, California institute of technology, 1990.

[54] P. Yeh, Introduction to photorefractive nonlinear optics, ch. 9, pp. 292–299. John Wiley & Sons, 1993.

[55] D. Psaltis, D. Brady, X. G. Gu, and S. Lin, "Holography in artificial neural networks," Nature, vol. 343, pp. 325–330, 1990.

[56] T. Merlet, D. Dolfi, and J.-P. Huignard, "A traveling fringes photodetector for microwave signals," IEEE Journal of Quantum Electronics, vol. 32, no. 5, pp. 778–783, 1996.

[57] A. V. Vanderlugt, "Adaptive optical processor," Applied Optics, vol. 21, no. , p. 4005, 1982.

[58] J. H. Hong and I. McMichael, "Novel optical technique for phased-array processing," Optical Engineering, vol. 30, no. 12, p. 1976, 1991.

[59] K. Wagner and D. Psaltis, "Multilayer optical learning networks," Applied Optics, vol. 26, pp. 5061–5076, 1987.

[60] M. A. Neifeld and D. Psaltis, "Optical implementations of radial basis classifiers," Applied Optics, vol. 32, pp. 1370–1379, 1993.

[61] K. Wagner and T. M. Slagle, "Optical competitive learning with VLSI liquid-crystal winner-take-all modulators," Applied Optics, vol. 32, pp. 1408–1435, 1993.

[62] T. Hara, Y. Ooi, Y. Suzuki, and M. H. Wu, "Transfer characteristics of the microchannel spatial light modulator," Applied Optics, vol. 28, pp. 4781–4786, 1989.

[63] J. Feinberg, "Assymetric self-defocusing of an optical beam from the photorefractive effect," J-Opt-Soc, vol. 72, pp. 46–51, 1982.

[78] A. V. Krishnamoorthy, J. E. Ford, K. W. Goossen, J. A. Walker, A. L. Lentine, S. P. Hui, B. Tseng, L. Chirovsky, R. Leibenguth, D. Kossives, D. Dahringer, L. A. Dasaro, F. E. Kiamilev, G. F. Aplin, R. G. Rozier, and D. Miller, "Photonic page buffer based on GaAs multiple-quantum-well modulators bonded directly over active silicon complementary-metal-oxide-semiconductor (CMOS)," Applied Optics, vol. 35, pp. 2439–2448, 1996.

[79] D. A. B. Miller, D. S. Chemla, T. C. Damen, A. C. Gossard, and W. Wiegmann, "Bandedge electro-absorption in quantum well structures: The quantum confined stark effect," Phys. Rev. Lett., vol. 53, pp. 2173–2177, 1984.

[80] D. A. B. Miller, D. S. Chemla, T. C. Damen, A. C. Gossard, and W. Wiegmann, "Electric field dependence of optical absorption near the bandgap of quantum well structures," Phys. Rev., vol. B32, p. 1043, 1985.

[81] Z. Gan, Acoustic Signal Processing in Photorefractive Optical Systems. PhD thesis, University of Colorado, 1994.

[82] D. Armitage, "Liquid-crystal display device fundamentals," in M. A. Karim, ed., (Electro-optical displays), ch. 2, pp. 19–67. Dekker, 1992.

[83] K. Y. Hsu, H. Y. Li, and D. Psaltis, "Holographic implementation of a fully connected neural network," Proceedings Of The IEEE, vol. 78, pp. 1637–1645, 1990.

[84] A. Brignon and K. H. Wagner, "Polarization state evolution and eigenmode switching in photorefractive BSO," Optics Communications, vol. 101, pp. 239–246, 1993.

[85] R. T. Weverka, K. Wagner, and A. Sarto, "Photorefractive processing for large adaptive phased-arrays," Applied Optics, vol. 35, pp. 1344–1366, 1996.

[86] A. W. Sarto, K. H. Wagner, R. T. Weverka, S. Weaver, and E. K. Walge, "Wide angular aperture holograms in photorefractive crystals by the use of orthogonally polarized write and read beams," Applied Optics, vol. 35, no. 29, pp. 5765–5775, 1996.

[87] A. Sarto, Adaptive Phased-array Processing using Photorefractive Volume Holograms. PhD thesis, University of Colorado, May 1996.

[88] G. Kriehn, P. E. X. Silveira, G. S. Pati, F. Schlottau, and K. H. Wagner, "BEAM-TAP RF-photonic adaptive-array processing," in Interactions between Microwaves and Optics, August 2000.

[89] G. Kriehn, G. S. Pati, P. E. X. Silveira, F. Schlottau, K. H. Wagner, and D. Dolfi, "Demonstration of optical beam forming using BEAMTAP," in Intern. workshop on photonics for antennas, September 2000.

[90] J. R. T. Compton, Adaptive Antennas. Prentice Hall, 1988.

[91] A. J. M. Kiruluta, G. Kriehn, P. E. X. Silveira, and K. H. Wagner, "Adaptive beamforming with TDI CCD based true-time-delay processing," in Proc. SPIE, Vol. 3804, SPIE, July 1999.

[92] P. E. X. Silveira, G. Kriehn, A. Kiruluta, S. Weaver, K. H. Wagner, and R. T. Weverka, "All-optical antenna array adaptive processing system," in Proc. SPIE, Vol. 3810, SPIE, July 1999.

[93] A. Yariv, Optical Electronics. Sounders College Pub, fourth ed., 1991.

[94] B. H. Kolner and D. W. dolff, "Intermodulation distortion and compression in an integrated electrooptic modulator," Applied Optics, vol. 26, no. 17, pp. 3676–3680, 1987.

[95] C. Bishop, Neural Networks for Pattern Recognition, ch. 9, pp. 338–340. Clarendon Press, 1995.

[96] P. S. R. Diniz, Adaptive Filtering: Algorithms and Practical Implementation, ch. 3, pp. 75–78. Kluwer Academic Publishers, 1997.

[97] P. E. X. Silveira, G. S. Pati, and K. H. Wagner, "Optoelectronic implementation of a 256-channel sonar adaptive array processor," to be published in Applied Optics.

[98] Motorola, Inc., M68HC11 Reference Manual, 1991.

[99] Motorola, Inc., M68HC11 E HCMOS Microcontrller unit, 1995.

[100] National Instruments Corp., NI-DAQ User Manual for PC Compatibles, September 1994.

[101] National Instruments Corp., LabVIEW User Manual, January 1998.

[102] National Semiconductor Corp., LM6121/LM6221/LM6321 High speed buffer, May 1998.

[103] Motorola, Inc., M68HC11EVBU Universal Evaluation Board User's Manual, April 1999.

[104] Mini-Circuits, RF/IF Designer's Handbook, 1992.

[105] G. L. Abbas, V. W. S. Chan, and T. K. Yee, "Local-oscillator excess noise supression for homodyne and heterodyne detection," Optics Letters, vol. 8, pp. 412–422, 1983.

[106] Burr-Brown Corp., High precision, low noise operational amplifiers, May 1999.

[107] National Semiconductor Corp., CLC532 High-speed 2:1 analog multiplexer, June 1999.

[108] Burr-Brown Corp., Ultra-low noise precision operational amplifiers, March 1998.

[109] S. Sakamoto and T. Yazaki, "Anomalous electro-optic properties of ferroelectric strontium barium niobate and their device applications," Appl. Phys. Lett., vol. 22, no. 9, pp. 429–431, 1973.

[110] S. MacCormack and J. Feinberg, "Revealing 180⁰ domains in ferroelectric crystals by photorefractive beam coupling," Appl-Optics, vol. 35, no. 30, pp. 5961–5963, 1996.

[111] S. M. V. Grubsky and J. Feinberg, "All-optical three-dimensional mapping of 180⁰ domains hidden in a BaTiO₃ crystal," Optics-Lett, vol. 21, no. 1, pp. 6–8, 1996.

[112] P. L. P. Mathey, P. Julien and D. Rytz, "Photorefractive detection of antiparallel ferroelectric domains in BaTiO₃ and BaTiO₃:Co crystals," Appl-Phys-A, vol. 66, pp. 511–514, 1998.

[113] P. Yeh, Introduction to photorefractive nonlinear optics, ch. 4, pp. 118–150. John Wiley & Sons, 1993.

[114] P. S. R. Diniz, Adaptive Filtering: Algorithms and Practical Implementation, ch. 4, pp. 150–153. Kluwer Academic Publishers, 1997.

[115] P. E. X. Silveira, G. S. Pati, and K. H. Wagner, "Optical adaptive array for source position estimation," in Proc. Int. Conf. on Optics in Computing, OSA, January 2001, pp. 27–29.

[116] Research Systems, Inc., IDL User's Guide, 1997.

Appendix A

Photorefractive grating formation and diffraction in FIRNN architectures

A.1 Introduction

In this appendix an operator-notation analysis for the simultaneous PR grating formation and readout using the input delay plane and output delay plane architectures is presented. It is shown that diffracted beams in conjugate planes can be produced by reading the PR grating with the fields propagating in the same directions as those used to record it. This allows the FIRNN architectures to simultaneously address the same grating in forward and backward propagation, as indicated by the multi-layer architectures presented in chapter 3. Figure A.1 shows a diagram used to explain the holographic grating formation and readout. Compare it to the architecture illustrating holographic interconnects between a 2-D plane and a 1-D vector (Fig. 3.6) and note that it consists of the same diagram seen from an angle perpendicular to the $z - x$ plane.

A.2 Analysis

Let $E^A(z'_0; t)$ and $E^B(x_0, y_0; t)$ represent the time-varying forward propagating signals whose interference pattern records a dynamic grating in the PR crystal. Their modulating SLMs (SLM 1 and SLM 2) and detectors (Detector array 1 and Detector array 2) are one-dimensional and two-dimensional, respectively. Note that such a setup is sufficient to describe all the FIRNN architectures presented in chapter 3 (input delay

SLM 2 is exactly imaged pixel by pixel onto detector array 2. The PR crystal is placed at a point of intersection between the two modulated beams, but it is in an arbitrary plane (not necessarily near an image plane or Fourier plane). To simplify the calculation of the field distributions at any arbitrary point inside the PR crystal, we use mathematical operators for free-space propagation and lens transformation [87]. Using the coordinate system indicated in figure A.1, the field distributions inside the PR crystal resulting from input fields E^A and E^B are given by

$$E^A(x,y,z;t) = \mathcal{D}_x^{(y,z;y_1,z_1)}\mathcal{L}_3^{(y_1,z_1;y_0,z_0')}\left\{E^A(z_0';t)\delta(y_0)\right\}. \tag{A.1}$$

$$E^B(x,y,z;t) = \mathcal{D}_z^{(x,y;x_1,y_1)}\mathcal{L}_1^{(x_1,y_1;x_0,y_0)}\left\{E^B(x_0,y_0;t)\right\}. \tag{A.2}$$

where $\mathcal{D}$ is the Fresnel operator with an index subscript representing the propagation distance and direction inside the crystal, and superscript representing the transformation of coordinates of the field. $\mathcal{L}$ is the lens transformation operator which describes the field propagation from SLM 2 to the lower crystal face ($\mathcal{L}_1$), from the lower crystal face to detector array 2 ($\mathcal{L}_2$), from SLM 1 to the leftmost crystal face ($\mathcal{L}_3$), and from the leftmost crystal face to detector array 2 ($\mathcal{L}_4$). Since the lens systems ($\mathcal{L}_1, \mathcal{L}_2$) and ($\mathcal{L}_3, \mathcal{L}_4$) image their input SLMs onto their respective detector arrays, they present the following properties:

$$\mathcal{L}_2^{(x_2,y_2;x_1,y_1)}\mathcal{L}_1^{(x_1,y_1;x_0,y_0)} \equiv \mathcal{I}_M^{(x_2,y_2;x_0,y_0)}, \tag{A.3}$$

$$\mathcal{L}_4^{(y_2,z_2';z_1,y_1)}\mathcal{L}_3^{(z_1,y_1;z_0',y_0)} \equiv \mathcal{I}_{M'}^{(y_2,z_2';z_0',y_0)}, \tag{A.4}$$

where $\mathcal{I}^{(x_2,y_2;x_0,y_0)}$ denotes the Imaging (or Identity) operator, transforming coordinate system (x_0, y_0) into the coordinate system (x_2, y_2), with the subscripts M and M' representing the magnification factors associated with each one of the lens systems. For simplicity, we will assume telescopic imaging, without quadratic phase factors. The

Figure A.1: Photorefractive grating formation and readout in FIRNN multi-layer architectures. SLM 1 is imaged pixel by pixel onto detector array 2 and SLM 2 is imaged pixel by pixel onto detector array 1. The dimension of the SLMs and the detectors (1-D or 2-D) vary according to which architecture is being analyzed (conventional or delayed feedback). However, the diagram and its analysis apply to both architectures.

plane, output delay plane and their combinations in forward propagation and simultaneous back-propagation) by selecting the signals applied to the SLMs. The possible consequences of Bragg degeneracy during readout under this geometry has already been discussed in section 3.3.1.

Note that SLM 1 is exactly imaged pixel by pixel onto detector array 1, and that

Fresnel operator $\mathcal{D}$ is defined as

$$\mathcal{D}_z^{(x',y':x,y)}\{E(x,y,z=0;t)\} = \frac{e^{i\frac{2\pi}{\lambda}z}}{i\lambda z}\int\int E(x,y,z=0;t)e^{i\frac{\pi}{\lambda z}[(x'-x)^2+(y'-y)^2]}\,dx\,dy,$$
(A.5)

where λ corresponds to the optical wavelength inside the PR medium. In the subsequent analysis, the following properties of the $\mathcal{D}$ operator are used

$$\mathcal{D}_z^{*(x,y:x',y')} \equiv \mathcal{D}_{-z}^{(x',y':x,y)}$$
(A.6)

$$\mathcal{D}_{-z}^{(x',y':x,y)}\mathcal{D}_z^{(x,y:x',y')} \equiv \mathcal{I}.$$
(A.7)

An analogous definition and set of properties hold for the operator $\mathcal{D}_x$, in the x-dimension. Now, the evolution of the index grating formed in the PR crystal resulting from the superposition of the interfering fields (Eqns. A.1 and A.2) can be described using a simple relaxation equation:

$$G(x,y,z;t') = \beta\int_{-\infty}^{t'}E^{A*}(x,y,z;t)E^B(x,y,z;t)e^{-\frac{(t'-t)}{\tau_{pr}}}\,dt$$
(A.8)

where β represents the sensitivity of the PR crystal and τ_{pr} denotes the PR time constant.

The diffracted field resulting from the read-out of the index grating by the incident field distribution E^A is given by the field distribution at the crystal $[E^A(x,y,z;t')]$ multiplied by the crystal grating $[G(x,y,z,t')]$ propagated back to the surface of the crystal (operator $\mathcal{D}_{-z}^{*(x,y:x,y)}$), bringing the field to a plane where it can be operated by one of the imaging operators. The resulting field is integrated over the z-dimension (the other two dimensions of the volume hologram are taken care of by propagation back to the (x_1,y_1) plane) and propagated until the Detector array 1 (operator $\mathcal{L}_2^{(x_2,y_2:x_1,y_1)}$). That is,

$$E_{df}(x_2,y_2;t') = \mathcal{L}_2^{(x_2,y_2:x_1,y_1)}\left[\int_0^{L_1}\mathcal{D}_{-z}^{(x_1,y_1:x,y)}\left\{E^A(x,y,z,t')\,G(x,y,z,t')\right\}dz\right],$$
(A.9)

where L_1 is the length of interaction along the z-dimension of the PR crystal. Substituting $G(x,y,z,t')$ with the grating equation (Eqn. A.8) and expanding the field E^A

using equation A.1, we have

$$E_{df}(x_2,y_2;t') = \beta\mathcal{L}_2^{(x_2,y_2:x_1,y_1)}\int_0^{L_1}\mathcal{D}_{-z}^{(x_1,y_1:x,y)}\left\{\left[\mathcal{D}_x^{(y,x:y_1,x_1)}\mathcal{C}_3^{(y_1,x:y_0,z_0)}\right]^*\left\{E^{A*}(z_0';t)\delta(y_0')\right\}\right.$$
$$\cdot\left\{E^A(z_0';t)\delta(y_0')\right\}\mathcal{C}_3^{(y_1,x:y_0,z_0)}\mathcal{C}_1^{(x_1,y_1:x_0,y_0)}$$
$$\cdot\left[\mathcal{D}_z^{(x,y:x_1,y_1)}\mathcal{C}_1^{(x_1,y_1:x_0,y_0)}\right]E^B(x_0,y_0;t)\left.e^{-\frac{(t'-t)}{\tau_{pr}}}\right\}dt\,dz.$$
(A.10)

Rearranging the operators and using the conjugate properties (Eqns. A.6 and A.7), we have that $\mathcal{D}_x^{(y,x:y_1,x_1)}\mathcal{C}_3^{(y_1,x:y_0,z_0)}\mathcal{C}_3^{*(y_0,z_0':y_1,x_1)}\mathcal{D}_x^{*(y_1,x_1:y,x)} \equiv \mathcal{D}_x^{(y,x:y_1,x_1)}\mathcal{I}\,\mathcal{D}_x^{(y_1,x_1:y,x)} \equiv \mathcal{I}$, giving us

$$E_{df}(x_2,y_2;t') = \beta\mathcal{L}_2^{(x_2,y_2:x_1,y_1)}\int_0^{L_1}\mathcal{D}_{-z}^{(x_1,y_1:x,y)}\left\{E^A(z_0';t)\right.$$
$$\cdot\left.\int_{-\infty}^{t'}E^{A*}(z_0';t)E^B(x,y,z;t)e^{-\frac{(t'-t)}{\tau_{pr}}}\right\}dt\,dz.$$
(A.11)

Expanding the field E^B using equation A.2, we have

$$E_{df}(x_2,y_2;t') = \beta\mathcal{L}_2^{(x_2,y_2:x_1,y_1)}\int_0^{L_1}\mathcal{D}_{-z}^{(x_1,y_1:x,y)}\left\{E^A(z_0';t)\int_{-\infty}^{t'}E^{A*}(z_0';t)\right.$$
$$\cdot\left[\mathcal{D}_z^{(x,y:x_1,y_1)}\mathcal{C}_1^{(x_1,y_1:x_0,y_0)}\right]\left\{E^B(x_0,y_0;t)\right\}\left.e^{-\frac{(t'-t)}{\tau_{pr}}}\right\}dt\,dz.$$
(A.12)

Note that a transverse integration across the intensity $|E^A(z_0')|^2$ or $|\mathcal{L}_3E^A|^2$ or $|\mathcal{D}_x\mathcal{L}_3E^A|^2$ is invariant to propagation via generalized Parseval's theorem, which will allow us to integrate across z_0 in place of z. Rearranging the operators and using the conjugate property of the operator $\mathcal{D}$ (Eqn. A.7), we have that $\mathcal{L}_2^{(x_2,y_2:x_1,y_1)} \equiv \mathcal{L}_2^{(x_2,y_2:x_1,y_1)}$ $\mathcal{L}_1^{(x_1,y_1:x_0,y_0)}\,\mathcal{I}\,\mathcal{L}_1^{(x_1,y_1:x_0,y_0)} \equiv \mathcal{L}_2^{(x_2,y_2:x_1,y_1)}\mathcal{L}_1^{(x_1,y_1:x_0,y_0)}$. Using the first property of the lens transformation operator (Eqn. A.3), we have that $\mathcal{L}_2^{(x_2,y_2:x_1,y_1)}\mathcal{L}_1^{(x_1,y_1:x_0,y_0)} \equiv \mathcal{I}_M^{(x_2,y_2:x_0,y_0)}$. Substituting, we finally have

$$E_{df}(x_2,y_2;t') = \beta\mathcal{I}_M^{(x_2,y_2:x_0,y_0)}\int_0^{W_1}E^A(z_0';t')\int_{-\infty}^{t'}E^{A*}(z_0';t)E^B(x_0,y_0;t)e^{-\frac{(t'-t)}{\tau_{pr}}}\,dt\,dz_0',$$
(A.13)

where a lossless imaging system is assumed and the integral is now performed over W_1, the width of SLM 1. Note that the fields $E^A(z_0'; t')$ and $E^{A*}(z_0'; t)$ are spatial conjugates of each other. As such, the direction of propagation of $E_{df}(x_2, y_2; t')$ is given by the direction of propagation of the only remaining field, that is, $E^B(x_2, y_2; t)$. Thus, we have shown that by recording a grating with $E^{A*}(z_0'; t)$ and $E^B(x_0, y_0; t)$ and reading it out with $E^A(x_0, y_0, z; t')$ we get a diffracted field in the direction of $E^B(x_2, y_2; t)$. Analogously, the diffracted field resulting from the read-out of the index grating by the incident field distribution E^B is given by

$$E_{db}(y_2', z_2'; t') = \mathcal{L}_4^{(y_2, z_2'; x_1, y_1)} \left[\int_0^{L_2} \mathcal{D}_{-x}^{(y_1, z_1; y, z)} \left\{ E^B(x, y, z, t') G^*(x, y, z, t') \right\} dx \right], \tag{A.14}$$

where L_2 is the length of interaction along the x-dimension. Substituting and applying the operator properties (canceling conjugate operators and using the imaging operation defined in Eqn. A.4), we find that the field reaching the 1-D detector array is given by

$$E_{db}(z_2'; t') = \beta \mathcal{I}_{M'}^{(y_2', z_2'; y_0, z_0')} \int_0^{W_{y_2}} \int_0^{W_{x_2}} E^B(x_0, y_0; t') \tag{A.15}$$
$$\int_{-\infty}^{t'} E^{B*}(x_0, y_0; t)\, E^A(z_0'; t)\, \delta(y_0')\, e^{-\frac{(t'-t)}{\tau_{pr}}}\, dt\, dx_0\, dy_0,$$

which is a field propagating in the same direction as $E^A(z_0'; t)$, where W_{x_2} and W_{y_2} are the widths of SLM 2 in the x and y dimensions, respectively.

Note that the fields represented by equations A.12 and A.15 are in the same form as the detected diffracted fields represented by equations 3.7, 3.14, 3.18, and 3.21.

Appendix B

BTSIM – Algorithmic simulator for adaptive antenna arrays

B.1 Introduction

This appendix presents a computer program developed for simulating different architectures for adaptive antenna arrays. Initially, it was developed with the BEAMTAP architecture in mind and, therefore, it was named BTSIM (for BEAMTAP Simulator). When the FIRONN architecture was designed, the simulator was extended in order to allow for the simulation of the conventional LMS algorithm. However, as explained in chapter 4, both architectures converge to the same set of weights, presenting equivalent performance at steady-state. This appendix also discusses some related algorithmic issues and their relevance to the simulations and to the actual implementation of the algorithm.

BTSIM, provides its users with the following advantages and capabilities:

(1) the capability to determine the effect that different parameters have on the performance of the array;

(2) the flexibility to experiment with different input signal scenarios and different algorithms;

(3) a source of input data which can be used in experimental implementations, such as the FIRONN architecture described in chapter 5.

Figure B.1: The startup window of the BTSIM program. The main menu bar, at the top of the window, evokes six different sub-menus. The top and bottom sub-windows are used for displaying simulated data. At startup, they show diagrams illustrating the BEAMTAP architecture.

B.2 Operation

The **BEAMTAP Simulator version 1.1** is an easy to use but powerful graphic user interface (GUI) driven program written under the IDL version 5.0.2 mathematical computer language [116]. Figure B.1 shows the program window at startup. The top and bottom windows are used for displaying simulated data and, at startup, they show diagrams illustrating the BEAMTAP architecture.

The top menu bar consists of six option buttons which, when clicked by a mouse button, evoke sub-menus, each with its own list of options. The operation of each sub-menu is described next.

B.2.1 File

This button displays four choices: 'Produce PS output', 'Save pars', 'Load pars' and 'Done'.

Produce PS output

This option allows the user to control which PostScript files to be generated and where. It has two sub-options: 'Select output path' and 'Produce PS'. The 'Select output path' sub-option prompts the user to select a directory where the PS files will be saved. The current path name is displayed along with a list of options. The user may cancel the operation, what keeps the current selection, or he/she may select a new path name. Notice that the path name must finish with a '/' character (the default), or else the program will issue a message and prompt the user for a new selection.

The 'Produce PS' sub-option evokes a widget (Fig. B.2) which lists the available printing options. They produce the same graphs that are viewed under the 'Display signals' option (explained below), except for 'Internal variables' option. This option produces images of internal variables, such as the final weights and the contents of the tapped-delay-lines, without axes. These are generally used by other programs, such as TGIF, in order to create more complex figures, such as the ones shown at the startup window. The 'Produce PS' widget also has 3 buttons: 'Create selected files', which allows the user to create only the PostScript files selected above; 'Create ALL files', which is equivalent to selecting all PostScript output files; and 'Cancel', which closes the widget, allowing the user to cancel the operation.

Save pars

• **Variables** – in order to save disk space, not all variables are saved. However, given the parameters and variables saved, the unsaved variables usually can be quickly reevaluated. Weights, input and output tap values and the calculated receptivity pattern are always stored. The input signal, desired signal and calculated fringe pattern are not. For a detailed description of each variable, see the file `btinit.pro`.

Load pars

This option, analogously to 'Save pars', prompts the user for a file name. The program will not allow the user to select a file that does not exist. The user may cancel the operation. The variables and parameters restored are the same as described above.

Notice that whenever new parameters are loaded, all current ones are lost. The current input signal, desired signal and calculated fringe pattern, however, are kept intact, but will be overwritten once they are recalculated. Once loading is complete, the program compares the current program version with the program version under which the loaded file was saved. In case of a mismatch a warning message is displayed.

Done

This option has two sub-options: 'Go to IDL' and 'Quit'. The first option closes all widgets and takes the user to the IDL prompt. All current variables remain in memory and usual IDL commands are accepted. The second option returns the user to the shell prompt after requesting confirmation. All unsaved variables and parameters are lost.

B.2.2 Display signals

This main menu item is used to calculate and display plots of different variables describing the state of the simulation and performance of the phased array. It has

Figure B.2: The 'Produce PS' widget, showing a selection list of possible plots which can be generated as PostScript output. Some selections, such as 'Input signal' and 'Final weights' produce more than one output file.

The 'Save pars' option allows the user to save the parameters and variables that have been created or modified during the current session. It provides the user with a sample path name and a file name which was derived from the computer system time at the moment when the program was evoked from the command line. This way a unique file name is automatically generated. The user may accept the suggested name, select a new one, or cancel the operation.

The 'Save pars' operation saves the following parameters and variables:

• **Parameters** – ALL of them (see section B.3 for more details), including the program version number, units used, system file names and system flags used to determine whether certain computations have already been performed for a given data set.

7 options: 'Input signal', 'Desired signal', 'Error & output', 'MSE & weight change', 'Final weights', 'Fringe pattern' and 'Receptivity pattern'.

Input signal

This option calculates and displays the input signal that will be detected by each of the antenna elements in the antenna array during simulation. Two images are displayed, one in real space and the other in Fourier space. In real space, the x-axis corresponds to space between antenna elements, variyng from 0 to L. The y-axis corresponds to simulated time, variyng from 0 to the maximum time (a parameter). In Fourier space the x-axis corresponds to temporal frequency and varies from 0 to one half the maximum sampling frequency (Nyquist). The y-axis corresponds to spatial frequency and varies from negative one over twice the inter-element spacing frequency and varies from negative to positive one over twice the inter-element spacing

a. Section B.2.3 explains how to modify the parameters that define the input signal.

Desired signal

This option calculates and displays the desired signal used by the adaptive algorithm. It is also the signal towards which the main lobe of the antenna's directivity pattern will be pointed during beamforming. Analogously to the 'Input signal' option, the desired signal is also displayed in real space and Fourier space.

Error & output

This option displays the error and output signals calculated during the BEAM-TAP simulation. The first one is entitled 'Learning curve' and displays the square of the norm of the simulation error vector, which measures how far the output signal is from the desired signal. The second one displays the output progression during the simulation.

In both graphs the x-axis represents simulated time. Notice that it may not

necessarily start from zero, since several simulations may take place over the same set of weights. The total simulation time is recorded and the graphs display the time of the last simulation performed.

MSE & weight change

This option displays two graphs: the progression of the mean squared error (MSE) from the first to the final repetition of the simulation is displayed in the main window, while the weight progression for the last simulation is displayed in a separate window. These graphs make it easier to determine whether the algorithm has already converged, besides giving insight into the dynamics of the convergence, such as the value of different eigenvalues, the influence of noise, etc.

Final Weights

This option displays two images representing the current state of the adaptive weights. The first one shows them in real space, where the x-axis represents the antenna array element, and the y-axis represents the taps connecting the weights to the output delay line. The second graph is a Fourier transform of the first, where the x-axis represents the corresponding spatial frequency of the input array elements, varying from -Nyquist to Nyquist, and the y-axis represents the corresponding temporal frequency of the delay line, varying from 0 to Nyquist.

Fringe pattern

This option displays the real part of the gratings that would represent the weights in a volume hologram. First the program prompts the user whether he wants to calculate (or recalculate) the fringe pattern. If so, the calculation (which could be somewhat slow) is performed and the result is displayed. If not, the previously calculated fringe pattern is displayed, if one already exists.

Receptivity pattern

This option is used to calculate and display different graphs representing the receptivity pattern of the antenna array. The receptivity pattern shows how sensitive is the array to signal at different frequencies and different angles of arrival.

There are two sub-options: 'Entire pattern' and 'Slices'. The first option calculates the directivity pattern from all angles, producing a 3-D plot. Its calculation is slow and, therefore, the user is prompted about whether he would like to perform it or not. If so, the calculation starts and the different angles are progressively displayed as they are used in the calculation, giving the user an idea of the duration of the calculation. If not, the previously calculated receptivity pattern is displayed, if one exists. Two graphs are displayed: one showing an image of the receptivity pattern and another showing a surface plot of the same information. In both graphs the x-axis represents the angle of arrival and the y-axis represents the temporal frequency. Figure B.3 shows how the 3-D receptivity pattern is actually displayed on the screen.

The second sub-option calculates the directivity pattern only at those angles at which a signal exists. These signals are defined in the 'Jammer signal pars' and 'Desired signal pars' option from the 'Modify parameters' main menu button. Therefore, signals can be included or excluded by modifying these parameters. That is, if the amplitude of a signal is defined to be different from zero, and the signal is in the far-field, the array directivity at the angle of arrival of the signal is calculated. This calculation is performed for all valid signals in the far-field and a 2-D plot showing all the 'slices' from different angles is displayed.

B.2.3   Modify parameters

This main menu button reveals seven different classes of parameters which may be altered: 'Array pars', 'Optic pars', 'Jammer signal pars', 'Desired signal pars', 'Algorithmic pars', and 'Display pars'. Each class is grouped in a separate widget parameters
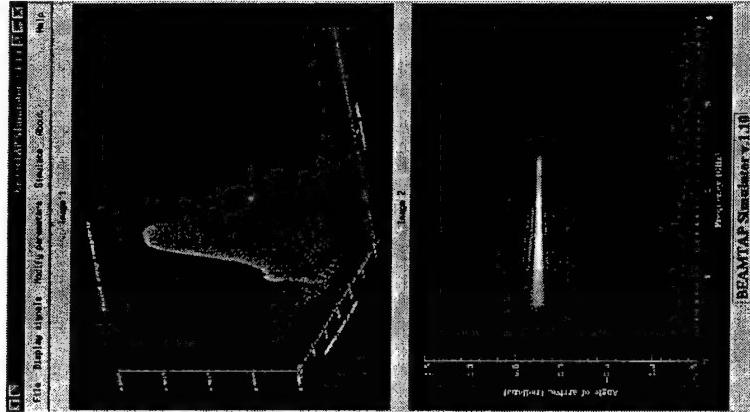
Figure B.3: Screenshot of the 3-D receptivity pattern as it is displayed in the program window. The image plot is shown on top and the surface plot is shown below.

that control different aspects of the simulation. The individual parameters are explained in detail in section B.3. The following is a general description of each class:

Array pars – these parameters control physical aspects regarding the phased array, such as the number of array elements and the number of taps in the delay line. It also allows the modification of the frequency/time units and the normalization constants (base frequency and wave velocity in the medium). Finally, it also displays calculated quantities which describe the expected performance of the array, such as the array fractional bandwidth and spatial resolution.
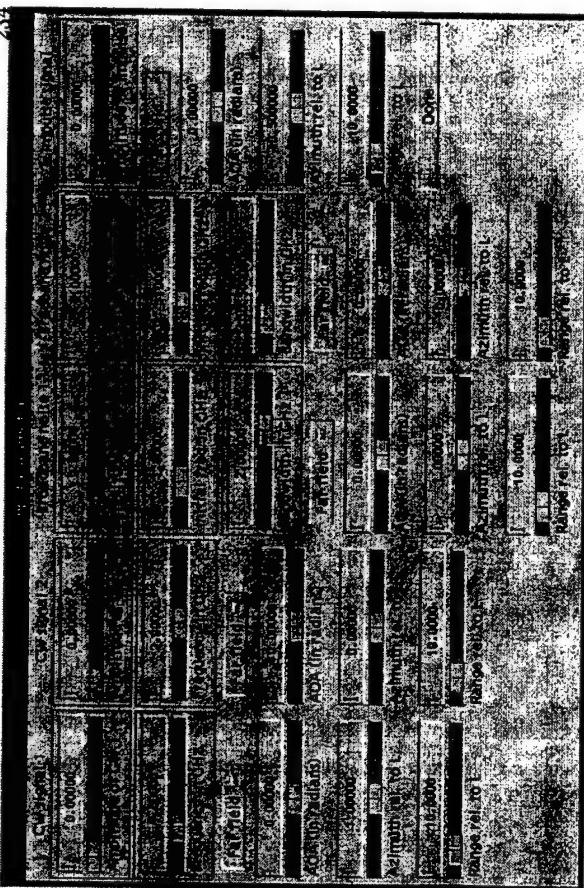
Figure B.4: Screenshot of the widget displayed when the 'Jammer signals pars' option is selected. A field can be modified by clicking and dragging the mouse, or by entering a new numeric value from the keyboard. Notice that if a jammer has zero amplitude, it is considered to be "turned off".

Optic pars – these parameters control the aspects governing the behavior of the optic elements in the simulation, such as the dimensions of the crystal composing the volume hologram, the wavelength of light and the non-linear parameters of the EO modulators.

Jammer signals pars – the parameters under this class determine the characteristics of the jammers present in the input signal. If a jammer has zero amplitude it is assumed that it is "turned off" and it is not calculated. Figure B.4 shows the widget used for changing these parameters. Multiple CW and broadband jammers may be simultaneously selected. Signals can be present in the far-field or the near-field of the antenna array. The maximum selectable frequency is limited by imposing Nyquist sampling of all signals.
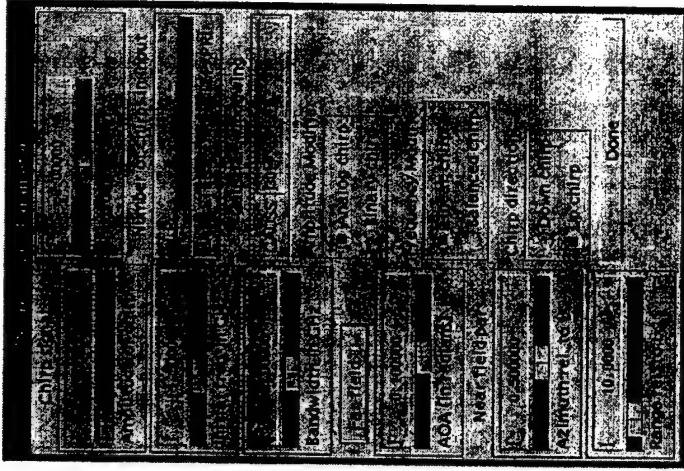
Figure B.5: Screenshot of the widget displayed when the 'Desired signal pars' option is selected. The chirp modifiers are selected by clicking one of the mutually exclusive buttons.

Desired signal pars – this class invokes a widget (Fig. B.5) which allows the modification of the parameters which select the characteristics of the desired signal, always assumed to be a chirp. Selecting a binary chirp actually defines the whole input signal to be binary quantized after all the jammers and the desired signal are added with their respective amplitudes, thus simulating modulation by a binary FLC-SLM.

Information signal pars – these parameters select the characteristics of a pseudo-random signal (the spread-spectrum 'information') which is added to the desired signal in the input, thus simulating the communication of information along with the desired signal. Selecting zero amplitude disables this feature.

Algorithmic pars – these parameters control the algorithmic aspects of the simulation, such as array sizes, sampling rate and gain factors. It also displays the total adaptation steps accumulated, a variable which is incremented as new simulations are performed over the same set of weights.

Display pars – these are the parameters responsible for tuning the appearance of the plots, such as color vs. gray scale, log plots vs. linear plots, etc.

B.2.4 Simulate

This main-menu button reveals five options: 'Run simulation', 'Nulling Test', 'Multiple jammer test', 'Reset variables' and 'Randomize weights'.

Run simulation

This option first prompts the user whether he/she would like to rerun the input signals (including the desired signal). This is done in order to be certain that the simulation is being performed over the latest version of these signals, after any possible change of parameters. However, it is kept optional in order to avoid possible lengthy recalculations.

What happens next depends on the settings selected under the 'Algorithmic pars'. If the user selected to 'Simulate field prop', field propagation is simulated, which uses complex evaluation of the weights and intermediate signals. The calculation becomes a lot slower, but it also reproduces more accurately an optical implementation. The fringe pattern is automatically calculated when field prop is selected.

If the user has selected 'Display intermediate results' under the Algorithmic pars, intermediate error plots are shown after a number of iterations. These plots provide the user with successive views of the instantaneous norm of the output error during the simulation. They satisfy the user's curiosity about the progression of the algorithm, but are not essential, since the final error plot is displayed at the end of that sequence of

the simulation.

The simulation will be run for as many repetitions as selected under the 'Algorithmic pars'. At the end of all repetitions the user is prompted about whether he/she would like to rerun the adaptation. If so, the same input and desired signals are used in the present state of the adapted weights. This allows the user to break a very long simulation in smaller ones, with a smaller number of time steps. Notice that the total number of adaptation steps is recorded, allowing the user to keep track of how many adaptations have already been performed over that set of weights.

After the simulation is over, the results can be displayed as explained in section B.2.2.

Nulling test

This option allows the user to determine the nulling performance of the antenna array under different scenarios. The complete nulling test consists of 3 steps:

(1) the user should run a long simulation in the presence of the desired signal only, until the array has satisfactorily adapted to that input environment. It is advisable that the current variables be saved at this point.

(2) then the desired jammers should be defined. This is done by selecting then under the 'Input signal pars' option. To select a jammer, just make sure that its amplitude is different from zero. The desired chirp, on the other hand, is assumed to be always present, with unit power. Obs.: it is generally necessary to reduce the learning rate after having added the jammers, in order to guarantee convergence in the presence of a stronger input signal.

(3) Run the nulling test.

At this point, the program will query the user for rerunning the input signals, which is recommended. Then the program will proceed in a loop consisting of two

major steps: 1) the determination of the array output due to the jammers alone, and 2) the adaptation of the array to the jammers and desired signals. This way the array performance in the presence of the jammers (the jammer nulling) can be traced over adaptation time. Four plots are produced between each repetition: 1) the output power due to the jammers alone versus the number of iterations. The jammer nulling at a given iteration is given by a ratio between the initial and a given data point in this plot. 2) the MSE of the adaptations versus the number of iterations. This plot allows the user to determine whether the adaptation is converging, how fast, etc. 3) the output of the array in the last repetition due to the jammers alone. This signal should approach zero as the jammers are nulled. 4) The central weight progression in the last repetition. This plot provides the user with an estimate of the adaptation dynamics and an insight on conversion time.

After the jammer output power has been calculated and the adaptation has been progressed for all the desired repetitions, the user is prompted about whether he/she would like to repeat the simulation, in which case adaptation and jammer power calculation continues from the point where it stopped.

Multiple jammer test

This selection automatically runs several consecutive simulations with an exponentially increasing number of broadband jammers, but with constant total jammer power. For each number of jammers, the weights are adapted and the SINR progression is recorded, allowing for the evaluation of the relation between the number of jammers and the dynamic behavior and final performance of the adaptive array processor.

Reset variables

If the user wishes to make sure that a simulation is starting from scratch, or if a simulation does not go as expected, it is possible to restart it by resetting the variables.

This option sets all weights to zero and resets the adaptation counter, the error and output signals and other internal variables (the output and error tap contents, and the flags indicating a previous calculation of the receptivity pattern and fringe pattern).

Randomize weights

This selection is identical to the previous one, except that the weights are set to small random values. It is useful to do this before the Nulling test is run, instead of resetting the weights to zero. Otherwise, the resulting zero output would not allow the nulling test results to be plotted in log scale for comparison.

B.2.5   About

This main-menu option displays a simple widget with information about the program, such as the program name and version number, name of the program designers and information about the place where the program was developed.

B.2.6   Help

This main-menu option displays two options: 'General help', which displays a file explaining the operation of the program, and 'List of pars', which displays a list of each of the simulation parameters divided by their classes, along with an explanation of each parameter.

B.3   List of parameters

The simulation parameters used by BTSIM are divided in six classes: Array parameters, Optic parameters, Jammer signal parameters, Desired signal parameters, Algorithmic parameters and Display parameters. These parameters control different aspects of the simulation, from signal specifications to the definition of the physical quantities involved. They can be modified by selecting the option 'Modify parameters'

from the main menu, as explained in section B.2. It is recommended that the Array and Optic parameters be defined first, followed by the Jammer and Desired signal parameters and the Algorithmic parameters, in that order. Each one of these set of parameters is explained next.

B.3.1 Array parameters

These parameters control physical aspects regarding the antenna array, such as the number of array elements and the number of taps in the delay line. They also include the frequency/time units and the normalization constants (base frequency and speed of propagation of the wave in the medium). Finally, they also include calculated quantities which describe the performance of the array. They are:

- **f** – the base frequency of the simulation. This parameter controls a multiplicative factor, with respect to which all frequency and time quantities are defined.

- **c-** speed of the wave in the medium. This parameter controls another multiplicative factor, with respect to which all length quantities are defined (in combination with the f parameter). At first it was used to define the speed of RF electromagnetic waves, but latter its definition was expanded in order to include sonar waves.

- **frequency unit** – the unit under which frequencies (and input signal times) are calculated. Choose among GHz, MHz or kHz.

- **Noise power** – the total power (in dBm) of the Gaussian zero mean white noise in the total signal detected by each antenna element at each time step.

- **Spatial appedization factor** – the exponential factor used in the spatial Gaussian appedization of the entire array. Set it to zero for no appedization.

- **rad** – the radius of the circular array. Set it to zero to define a planar array.

- **X_TALK** – the amount of cross-talk between a given antenna element and all of its neighbors. The cross-talk exponentially decreases with physical distance. Set it to zero for no cross-talk, or to one for complete cross-talk (all elements detect the same signal).

- **n** – index of refraction of the medium over which the input signal propagates. This is used in order to calculate the signal propagation delay from one antenna element to the next. Alternatively, this parameter along with **c** determine the speed of propagation of the signal wave in the medium.

- **theta_max** – the maximum angle of incidence of signals in the array. This parameter affects the spatial array resolvability and the maximum angle shown in the receptivity patterns.

- **NEl** – the number of array elements in the phased array antenna, expressed as a power of two. This parameter controls one of the dimensions of the weight matrix. Therefore, modifying this parameter resets the weight matrix.

- **TAPS** – the number of taps in the input and output delay lines, expressed as a power of two. This parameter also controls one of the dimensions of the weight matrix. Therefore, modifying this parameters resets the weight matrix.

The following variables are calculated from other parameters and their values are displayed in this widget:

- **B** – total signal bandwidth. Derived from the Jammer and Desired signal parameters.

- **Center freq.** – the center frequency of the desired signal.

- **F** – fractional bandwidth: the ratio between the total signal bandwidth and the desired signal center frequency.

- **Signal wavelength** – the wavelength of the center frequency of the desired signal.

- **Element spacing** – the spacing between array elements, set in order to Nyquist sample the input signal.

- **L** – total array length = a.NEl

- **Spatial array resolvability** – the minimum spatial wavelength of a far-field input signal incoming at an angle theta_max that can be resolved by the array.

B.3.2 Optical parameters

These parameters control the aspects governing the behavior of the optic elements in the simulation, such as the dimensions of the crystal composing the volume hologram. They also control the optical normalization quantities, such as the optical wavelength and index of refraction. The parameters are:

- **lambda_opt** – the optical wavelength. This parameter is a normalization constant used to scale other optical distances, such as the crystal dimensions.

- **n_opt** – the optical index of refraction. This is the average index of refraction inside the volume hologram.

- **Lx_xtal** – the x dimension of the crystal in terms of lambda_opt. This dimension corresponds to the NEl (spatial) dimension of the weight matrix. See 'Array parameters' for more information.

- **Lz_xtal** – the z dimension of the crystal in terms of lambda_opt. This dimension corresponds to the TAPS (temporal) dimension of the weight matrix. See 'Array parameters' for more information.

- **Vmod** – voltage amplitude of the EO modulator.

- **Vpi** – voltage V_π of the EO modulator (See section 4.2.6 for more information).

- **bp** – amplitude of uniformly distributed bias voltage variation in EO modulators across antenna elements. Set it to zero to prevent bias voltage from varying from its optimal point.

- **DECAY** – the photorefractive decay. This parameters controls a multiplicative constant by which all weights are multiplied at every iteration, being able to simulate a simple version of the photorefractive decay. If set to one, no decay is present. In practice, this parameter should be set to a value very close to one.

- **Lambda unit** – the unit of the lambda_opt distance. Choose among nm, μm or mm.

B.3.3 Jammer signal parameters

These parameters control the characteristics of the input jammer signal. Five simultaneous jammers can be selected (2 narrow band jammers, 2 broadband jammer and one broadband impulse jammer). See 'Multiple jammer test' in section B.2.4 for instructions on how to generate a larger number of jammers. The parameters are:

- **Amplitude of CW1, amplitude of CW2, etc.** - the amplitude of a given jamming signal. Set it to zero in order to disable a given jammer. The time required to generate the input signals is reduced as signals are turned off.

- **Frequency of CW 1, frequency of CW 2, initial freq, etc.** - the frequency (or initial freq.) of the corresponding signal. Notice that this number is limited to one half of the sampling frequency, making sure that Nyquist sampling is always respected. In the CW signals, this parameter is quantized in order to make sure that the sine waves represented comprise an integer number of periods. This allows the signal to be "wrapped" around and replayed over

multiple repetitions. In the broadband signals, changing this parameter causes the widget window to blink. This is because maximum bandwidths are being recalculated with respect to the Nyquist frequency as the initial frequency is modified.

- **Bandwidth of broadband noise 1, Bandwidth of broadband noise 2, etc.** - the bandwidth of the signal. Notice that initial frequency plus bandwidth is limited to one half of the sampling frequency, in order to prevent aliasing. The impulse signal is always broadband, with frequencies ranging from zero to Nyquist.

- **Near-field, far-field** – a flag indicating whether the signal is in the near-field or in the far-field.

- **AOA** – the angle of arrival of the signal, in radians, varying from -theta_max to theta_max (defined under 'Array parameters'). Notice that this parameter only affects far-field signals.

- **Azimuth relativel to L** – the distance from the source to a line perpendicular to the phased array, passing through its center point. L is the total array length. Notice that this parameter only affects near-field signals.

- **Range relative to L** – the distance from the source to a line passing through all points of the array. L is the total array length. Notice that this parameter only affects near-field signals.

B.3.4   Desired signal parameters

These parameters control the characteristics of the desired chirp signal. The parameters are:

- **Amplitude of the chirp** – Set it to zero in order to disable the desired signal.

- **Initial freq.** – the initial frequency of the broadband chirp. Again, it is limited to one half of the sampling frequency. Changing this parameter causes the widget window to blink. This is because the maximum bandwidth is being recalculated with respect to the Nyquist frequency as the initial frequency is modified.

- **Bandwidth** – the bandwidth of the signal. Notice that Initial frequency plus bandwidth is limited to one half of the sampling frequency, in order to prevent aliasing.

- **Near-field, far-field** – a flag indicating whether the signal is in the near-field or in the far-field.

- **AOA** – the angle of arrival of the signal, in radians, varying from -theta_max to theta_max (defined under 'Array parameters'). Notice that this parameter only affects a chirp in the far-field.

- **Azimuth relative to L** – the distance from the source to a line perpendicular to the phased array, passing through its center point. L is the total array length. Notice that this parameter only affects a chirp in the near-field.

- **Range relative to L** – the distance from the source to a line passing through all points of the array. L is the total array length. Notice that this parameter only affects a chirp in the near-field.

- **Chirp duration** – reducing the chirp duration increases the dead time between consecutive chirps.

- **Number of chirps** – the pulse repetition interval (PRI) is modified in order to accommodate the required number of chirps in a total of *MAX_STEPS* samples.

- **TIME_STEP** – the number of samples used to represent the base frequency. The minimum allowed value is 2. The maximum input frequency is determined by this number, since Nyquist sampling is enforced. Thus, the maximum allowed input frequency is given by $f/(2.TIME\_STEP)$, where f is defined under the 'Array parameters'.

- **num_repet** – the number of times the simulation is to be repeated. This allows the execution of very long simulations without using an extreme amount of memory.

- **DDEL** – the number of steps before the beginning of the desired signal. This is used in order to delay the desired signal with respect to the input signal and shift the position the cross-correlation pattern present in the weight matrix.

- **PAD** – the number of zeros padded to the input signal before its Fourier transform is performed, represented as a power of two. More zeros make the transform slower, but provide better resolved images of the input signal spectrum.

- **AOAS** – number of angles of arrival used in the calculation of the directivity pattern. This calculation is remarkably slow for large weight matrices. A small value of AOAS speeds up the calculation, but produces coarser receptivity patterns.

- **FRSIZEX** – size of the array used to represent the fringe pattern in the x dimension. Larger values make the fringe pattern calculation slower, but produce nicer images.

- **FRSIZEZ** – size of the array used to represent the fringe pattern in the z dimension. Larger values make the fringe pattern calculation slower, but produce nicer images.

- **Gaussian windowing** – Gaussian appedization of the chirp, to prevent leakage of high frequency components in the chirp spectrum. Set it to zero for no appedization.

- **Amplitude modifier** – a set of mutually exclusive buttons. If 'Binary chirp' is selected, the input signal is binary quantized after all jammers and desired signal are calculated.

- **Frequency modifier** – a set of mutually exclusive buttons. 'Linear chirp' selects a chirp with a frequency that increases linearly with time. 'Balanced chirp' selects a chirp with a frequency that increases as the power of 1.5 over time, producing a chirp with the same amount of energy at every frequency component.

- **Chirp direction** – a set of mutually exclusive buttons, selecting either a down chirp (frequency decreases with time) or an up chirp (frequency increases with time).

B.3.5   Algorithmic parameters

These parameters control the algorithmic aspects of the simulation, such as array sizes, signal sampling and gain factors. It also displays the total adaptation steps accumulated, a variable which is incremented as new simulations are performed over the same set of weights. The parameters are:

- **MAX_STEPS** – the total number of steps used in the simulation. It affects the size of the arrays used by the input and desired signals, error and output signals, etc. In order to save computational time, this parameter is represented as a power of two. The time required for signal generation can be greatly increased by increasing this parameter.

- **g1** – output feedback factor. A multiplication factor determining how much of the output will be applied towards feedback. Set it to zero to turn off output feedback.

- **g2** – error feedback factor. A multiplication factor determining how much of the error (or desired signal, in case of no output feedback) will be applied towards weight adaptation. It is the same as the learning rate when g1 = 1. Set it to zero in order to produce output with no weight adaptation.

- **Selected algorithm** – three mutually exclusive buttons select one of the three possible algorithms for simulation: Beamtap, Complex Beamtap and LMS. See 'Run simulation' in section B.2.4 for more details.

- **num-jam** – the number of jammers, in powers of two, to be used in the multiple jammer simulation.

- **num_adapt** – displayed at the bottom of the widget, indicating how many adaptation steps have already been applied to the current weights.

B.3.6 Display parameters

These parameters control some of the display options. They are:

- **BKGND** – amount of offset to be added to logarithmically scaled images. More offset makes most prominent features more visible, but hides less prominent background features.

- **disp_flag** – a flag controlled by a button. When set, intermediate plots of the learning error are displayed during the simulation.

- **Log scale in input plot** – when set, the image of the spectrum of the input signal is shown in log scale. Linear scale is used otherwise.

- **Plot image maps in color** – when set, image maps are plotted in color. Gray scale is used otherwise.

B.4 Program description

BTSIM is written as a modular collection of IDL procedures and functions. The program module `bt.idl`, organizes the common variable blocks and invokes the next module, `btinit.pro`, which initializes all variables and invokes the main menu widget procedure, `menu.pro`. The standard widget routines of IDL 5.0.2 are used, which require two procedures for defining a widget: the first procedure (*e.g.* `foo.pro`) defines the widget itself, including sub-widgets, geometry, initial values, etc. The second procedure (`foo.event.pro`) defines the actions that take place when the widget returns an event. Therefore, every widget will have 2 procedures associated with it. See reference [116] for more details.

B.4.1 List of procedures

The following is a list of the main modules used by the BTSIM program, followed by a general description of their function. Refer to the files themselves for additional programming comments.

- `bt.idl` – this file starts the program, defining the common variable blocks, invoking `btinit.pro` and the main menu widget. Common blocks are used in order to pass a large number of variables over several different routines, and to limit the variables that need to be saved. Once they are defined they can be recalled any time from within any procedure by simply using the command `common name`, where **name** is the name of the common block. However, common blocks must be defined only once. `bt.idl` lists all the variables in nine separate groups and keeps record of their names in their respective variable group string.

Then, these strings are used in order to define the common blocks. This way a new parameter can be easily added by simply adding its name to its respective group. The programmer does not have to worry about passing the new variable to the procedures that require it, since if a procedure requires the new variable it is most likely that it will already be importing its block. Also, the variable will be correctly saved or not, depending on whether its block is saved. Up to version 1.1, the only blocks which are not saved are `sig`, which contains the desired and input signals, and `calc`, which contains the fringe pattern. The variables contained by these blocks generally occupy a large amount of memory but they can be calculated without much delay.

- `btinit.pro` – this procedure defines all program variables before their first use. This assures that the program will start in a well behaved state. This file is useful to programmers and users who wish to manipulate variables in the IDL mode, since it lists most of the program variables with descriptive comments and a well behaved initial value.

- `menu.pro`, `menu_even.pro` – these procedures invoke the main menu widget and open the two display windows.

- `pschange.pro` – procedure used to select a new path for PostScript output.

- `psout.pro`, `psout_event.pro` – widget for selecting PostScript outut files to be produced.

- `prodps.pro` – procedure for creating selected PostScript files.

- `loadpars.pro`, `savepars.pro` – procedures responsible for load and save operations.

- `gotoidl.pro` – procedure used to quit the program, staying in IDL. No variables are lost.

- `quit.pro` – procedure used for quitting the program and IDL. All unsaved variables are lost.

- `dispd.pro`, `disps.pro` – procedures for displaying the desired signal and input signal, respectively.

- `disprec.pro`, `disprec_sl.pro`, `dispfr.pro` – procedures for displaying the total receptivity pattern, slices of the receptivity pattern and the fringe pattern.

- `dispeo.pro`, `dispw.pro`, `disptran.pro` – procedures for displaying the error and output plots, weight matrix images, and MSE and central weight transition plots.

- `probebt.pro` – procedure used to probe the adaptive array, generating the receptivity pattern.

- `dgen.pro`, `siggen.pro` – procedures for generating desired and input signals. They invoke the `*_array.pro` procedures.

- `fftsgen.pro` – procedure for calculating the averaged FFT of the input signal.

- `chirp_array.pro`, `delta_array.pro`, `gauss_array`, `sine_array.pro`, `noise_array.pro` – procedures used to generate arrays of delayed chirps, unit impulses, gaussian pulses, sine waves or filtered noise waves, respectively.

- `farfield.pro`, `nearfield.pro` – procedures used to calculate the waveform delays in the near-field and far-field.

- `chirp.pro`, `gauss_win.pro` – procedures for generating chirps and gaussian windows.

- `rfpars.pro`, `rfpars_event.pro`, `optpars.pro`, `optpars_event.pro`, `inppars.pro`, `inppars_event.pro`, `algpars.pro`, `algpars_event.pro` – wid-

gets for modifying the array, optical, input signals and algorithmic parameters, respectively.

- `rfcalc.pro`, `optcalc.pro` – procedures for calculating array and optical parameters, respectively.

- `btout.pro` – procedure for producing the regular BEAMTAP simulation and displaying some of the results.

- `beamtap.pro`, `comp_beamtap.pro` – procedure that performs the beamtap adaptation algorithm. Regular and complex version.

- `lms.pro` – procedure that performs the LMS algorithm.

- `forwprop.pro`, `comp_forwprop.pro` – procedures used to forward propagate an input through the adaptive array.

- `nulltest.pro` – procedure used to monitor dynamic jammer nulling.

- `resetw.pro` – procedure used the reset program variables and control flags.

- `power.pro` – procedure used to calculate the power in a signal.

- `aboutbt.pro` – displays general information about the program.

- `help.txt`, `helpars.txt` – text help files invoked from the main menu.

- `btsim1.jpg`, `btsim2.jpg` – JPEG images shown at initialization.

This thesis entitled:
Developments in Photorefractive Two-Beam Coupling Systems
written by Valéria B. Damião
has been approved for the
Department of Electrical & Computer Enginneering

_____
Dr. Dana Z. Anderson

_____
Dr. Kelvin Wagner

Date_____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards fo scholarly work in the above mentioned discipline.

DEVELOPMENTS IN PHOTOREFRACTIVE

TWO-BEAM COUPLING SYSTEMS

by

Valéria B. Damião

B.A., University of Brasília, Brazil, 1990

M.S., University of Colorado, USA, 1995

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Electrical & Computer Engineering

2000

Damião, Valéria B. (Ph.D. Electrical Engineering)

Developments in Photorefractive Two-Beam Coupling Systems.

Thesis directed by Dr. Dana Z. Anderson

Photorefractive systems hold the promise for improving the performance of photonic signal processing applications. This thesis describes several developments pertaining to such systems, in particular to those employing two-beam coupling.

The utility of photorefractive systems is stymied by the lack of a component-oriented technology. Conventionally, an entire optical table, full of mirror mounts and other optical components, is required to build a photorefractive system. To address this shortfall, two-beam coupling modules were designed and fabricated, whose ports were standard multimode fiber optic connectors instead of free-space propagating beams, dispensing with the need for time-consuming alignment and adjustments. These modules employ lenses to couple light into and out of our chosen photorefractive medium, crystalline barium titanate. We demonstrate the utility of the modular concept by implementing an autotuning filter constructed solely by these modules.

For even greater integration of the two-beam coupling units, we present a novel spherical crystal geometry which incorporates the functions of both the lenses and of the photorefractive medium in a single element. While investigating these spherical crystals, we observed internal whispering-gallery mode oscillations. These oscillating patterns adopted simple polygonal shapes depending on the pump entry and crystal-axis angles. We modeled the occurrence conditions of the lowest order triangle pattern by assuming a single two-beam coupling interaction, using standard plane-wave theory.

Also, as part of a higher-level photonic system, we required a carrier suppressor to remove unwanted correlation between different signal-containing beams. By use of two-beam coupling in the novelty-filter configuration, we obtained more than 70 dB of carrier suppression, to our knowledge, the highest reported in the literature. The use of a simplified, geometrical interpretation of an operator theory for two-beam coupling, which provided the proper choice of parameters needed for perfect suppression, was crucial in obtaining this result.

Finally, the full formalism of the operator theory for two-beam coupling is introduced. This formalism provides the freedom of representing the information to best suit the information-processing problem at hand. A closed-form solution for complex coupling is presented for the case of a single spatial mode interaction, the "2-by-2" case. The already reported solutions for pure-real coupling, corresponding to energy transfer only, and that for pure-imaginary coupling, phase transfer only, are given as special cases of the 2-by-2 complex coupling solution.

ACKNOWLEDGMENTS

I gratefully acknowledge the following:

First, Prof. Dana Anderson, for the opportunity to work on a well equipped laboratory and for the financial support (with the exception of the last semester) without which this Ph.D. would not have been possible. I am grateful for the most salient, eye-opening, of his teachings: the essentiality of socio-political and communication skills for a successful career.

Most of all, Bruce Tiemann, my partner and unofficial experimental advisor, for his dedication in teaching me the workings of nature in so many different aspects and for helping me develop a set of laboratory skills of fundamental importance on the work developed in this thesis.

Specially, Vladimir Schkunov and Ben Thompson, for their patience in elucidating numerous theoretical questions, and Prof. Kelvin Wagner for kindly listening and providing academic counseling.

Finally, all colleagues who participated in one way or another in my learning experience. In particular, Leslie Czaia, for preparing several of the crystals used in our experiments, for patiently sharing her talents in crystal processing techniques, and for guiding us through the hidden nooks of our laboratories.

A meus queridos pais, pela constante e incansável dedicação.

vi

CONTENTS

FIGURES

CHAPTER 1:
INTRODUCTION

1.1. MOTIVATION

The inherently parallel nature of optical information processing, coupled with its typically higher bandwidth, confers a potential advantage over conventional electronically-based processing, such as digital signal processing. Photorefractive media store real-time holograms which have memory and can provide high optical gain (20 cm^{-1}) at moderately low intensities on the order of W/cm^2, characteristics which are useful in the implementation of optical information processing architectures. Photorefractive-based systems can be subdivided in three main categories: optical computing, image processing and signal processing. Examples of optical computing systems include associative memories and correlators [Gabor, '69; Khoury, '94; Neifeld, '93; Staebler, '75; Yu, '94], neural networks [Denz, '99; Gu, '93; Psaltis, '88], and reconfigurable array interconnects [Anderson, '87; Marrakchi, '90; Weiss, '89]. Image processing systems include functions such as image amplification [Hong, '90], image thresholding [Klein, '86a; Sayano, '88], edge enhancement [Feinberg, '80; Joseph, '92], amplifying spatial filtering [Chang, '90; Fainman, '86], novelty filters [Anderson, '89; Khoury, '91; Uesu, '95]. Narrow-band filters [Hong, '93; Rakuljic, '93] and optical heterodyning [Hamel de Monchenault, '88; Khoury, '93] are examples of signal processing systems. Photorefractive systems applications have been reviewed in [Gunter, '88].

The emphasis in our research group is to perform information processing by use of photorefractive oscillating architectures which we call "rings", such as the flip-flop [Anderson, '91], the bistable ring [Lininger, '90], the demultiplexer [Saffman, '91], and the feature extractor [Anderson, '92]. These demonstrations are described as "self-organized", for the reasons that no external programming of the photorefractive ring is needed, nor is any *a priori* information about the input signals, except for the requirement that they be both temporally and spatially orthogonal. The dynamics of a ring oscillator generate real-time volume holograms in the photorefractive crystals which we call gratings. The competition dynamics between the generated gratings allow for different algorithms, such as separating temporally uncorrelated signals in different rings in the demultiplexer, or extracting the principal component within the input signals in the feature extractor.

This thesis focuses upon a fundamental functional unit in photorefractive processing: the two-beam coupling interaction. It is the main functional unit from which the ring systems are built, and it is also used in many of the more general applications mentioned above, such as in image amplifiers, novelty filters, notch filters, and others.

The next couple chapters in this thesis are motivated by the importance of developing a more component oriented technology for photorefractive functions, as it will significantly contribute to making photorefractive system testing and building more practical. Its widespread use in photorefractive systems justifies the choice of two-beam coupling as the example-function for testing such technology. Naturally, the same concept can be used for other photorefractive functions as well, such as

phase conjugation or four-wave mixing. The role that the development of such a component technology could represent to photorefractive systems can be illustrated by drawing an analogy with electronic systems, where systems can be assembled by the use of chips with a wide variety of function, canceling the need of building the system from the transistor level. For this purpose, in chapter 2, we introduce a modular design, where each module is pre-aligned to perform a particular photorefractive function, and the modules are interconnected by multimode optical fibers. Such technology allows systems containing several modules to be easily assembled and re-configured without the need of any system alignment. Each module is built for a specific desired function – a two-beam coupling module built for gain will be optimized differently than one built for novelty filtering. In the end of chapter 2, we give an application example using such modules. There we show how their use allowed a quick and easy verification of the change in behavior of a ring system when an extra module is inserted into its feedback loop, sparing the need for tedious realignment on the conventional optical table.

In chapter 3, we introduce the use of spherical photorefractive crystals. Their original purpose was in developing more compact two-beam coupling units. Due to the curvature and high refractive index of the crystal, no other optical elements are necessary for these modules except for the input and output fibers and the crystal itself. A built example of such module is shown herein. Interestingly, when testing the spherical crystals, we found that a single pump beam generated oscillating patterns within the crystal. These patterns arise from total internal reflections within the crystal, forming closed-path oscillating modes, known as whispering-gallery

modes. A spherical disk, an equatorial slice of the sphere containing the crystal's uniaxis, the "c-axis", was produced to allow better visualization of the patterns. The occurrence of the strongest mode, the triangle pattern, was mapped as a function of pump input and c-axis angles. A simple theory, assuming pumping of the modes by means of a single two-beam coupling interaction, resulted in good qualitative agreement with the experimentally obtained mapping. Due to the lack of a phase conjugate of the pump-beam, we believe that the oscillations are unidirectional, meaning that these structures are miniature analogous of unidirectional ring oscillators pumped by a two-beam coupling interaction, which is the architecture used for several of our signal processing systems, such as the feature extractor.

As mentioned above, the ring processors developed in our laboratories typically require that the input signals be temporally and spatially orthogonal. However, when using electrooptic modulators to impose a temporal signal upon different optical beams, the signals in these beams will have a correlation term due to the common carrier. Therefore, for the ring processor to function properly, it becomes necessary to remove the optical carrier from each signal beam. This is the case for a system being currently developed, the "optically-smart antenna array", which we describe in more detail in chapter 4. This system contains a feature extractor, currently renamed as the "autotuning filter", whose purpose is to select the strongest signal within the several signals presented at its input. Because the optical signals in this system are generated by using an electrooptic modulator, for the reason just explained above, a carrier suppressor is needed in front of the filter's input. Once again, two-beam coupling proves its versatility by performing adaptive carrier

suppression when used in a novelty-filter configuration. Carrier suppression of more then 70 dB is demonstrated. Such results were possible due to the development of an operator formalism which is the subject of the last chapter. In chapter 4, we provide a simplified, geometrical picture of this operator theory applied to the carrier suppression problem. This geometrical picture provided a greater intuitive understanding of the suppression mechanism, and allowed us to set the parameters involved in the carrier suppressor, such as input intensity ratio of the beams and the modulation strength.

Finally, the last chapter describes the operator algebra formulation for two-beam coupling. In 1979, Kukhtarev published a theoretical model for beam coupling which employed coupled wave theory [Kogelnik, '69] taking into account the effect of self-diffraction of light induced gratings. [Kukhtarev, '79] The equations presented in the next section are based on the plane-wave approximation to this theory, considered the foundation for two-beam coupling modeling. However, as the number of plane-waves interfering in the crystal increases, this theory becomes increasingly more difficult and awkward to deal with. A generalized operator algebraic theory, for which the input beams can be cast not only on a plane-wave basis, but on any set of orthogonal spatial bases is then indicated. The operator theory presented in chapter 5 treats the photorefractive medium as a black box described by a coupling operator T, which acts upon an input field vector to produce the output field vector, $E(out)=T E(in)$. The field vectors are represented in whichever basis is better suited to the information processing task at hand. Due to this freedom in representing the information, the operator theory bridges a gap between the physics of two-beam

coupling and its theoretical-computational representation, leading to a deeper intuitive understanding of the beam coupling phenomena. In fact, this is exemplified in chapter 4, section 4.4, where a geometrical picture of the operator formalism led to the prediction of carrier suppression features by simply graphically representing the field vectors and studying the action of T upon these vectors. In the last chapter, we present the analytical solution for the coupling matrix T for the case of one spatial mode in each of the two input beams. We present the "complex-coupling" solution, which corresponds to the general case consisting of a combination of intensity coupling (real case) and phase coupling (imaginary case).

Before proceeding to chapter 2, in the next section we present a brief summary of two-beam coupling, for those readers who are not familiar with the process.

1.2. TWO-BEAM COUPLING: AN OVERVIEW

This overview is just that, an overview, where an effort is made to only cover the main aspects of two-beam coupling necessary for the understanding of the following chapters. We explain the physical phenomenon behind the two-beam coupling process and introduce the two-beam coupling equations. For more detailed accounts we refer the reader to [Solymar, '96; Yeh, '93].

Two-beam coupling is a nonlinear optical interaction that can transfer energy or phase information between two overlapping beams in a photorefractive medium.

The two-beam coupling medium of this thesis is always an electrically unbiased, open-circuit, barium titanate (BaTiO$_3$) crystal, where the charge transport is primarily due to diffusion, resulting in a pure energy transfer process [Buse, '97]. Barium titanate displays high two-beam coupling gain over a wide wavelength range. (Barium titanate is a perovskite ferroelectric material belonging to the 4mm crystallographic group symmetry. It's photorefractive properties are widely available in the literature. [Klein, '86b; Mazur, '99].)

A two-beam coupling diagram is shown in figure 1.1. Two beams cross inside a crystal forming an interference pattern, which is sinusoidal if the beams are assumed to be plane waves. The bright regions of the interference pattern can excite charge carriers from impurity donors, typically iron ions (Fe^{2+} and Fe^{3+}) in barium titanate, to the conduction band. In this model we assume the only charge carriers are electrons. The electrons in the conduction band are free to migrate whereas the ionized donors are fixed to the crystal lattice. Electrons may experience several transitions as they migrate, in other words, they are re-combined with the ionized donors and re-excited to the conduction band. Because the excitation of the electron is proportional to the light intensity, more electrons are generated in the bright regions as opposed to the dark regions of the interference pattern. This causes an average migration or diffusion of electrons to the dark regions. This charge separation results in a charge density function (ρ) with the same periodicity of the interference pattern, as shown in figure 1.2. The charge density and the resulting space-charge field (E_{sc}) are related by Poisson's equation, $\nabla \cdot \varepsilon\, E_{sc} = \rho$, which introduces a 90° phase shift between ρ and E_{sc}.

Barium titanate displays a strong linear electrooptic effect, also known as Pockel's effect, which generates a refraction index grating in the medium which mimics the space-charge field. The optical fields which originally recorded this grating will now scatter from it, i.e., the fields experience self-diffraction. The 90° phase shift between the index grating and the interference pattern causes the diffracted beams to interfere constructively in one output direction (along the "gain beam") and destructively in the other output direction (along the "loss beam"). This results in an energy transfer between the beams. The direction of energy transfer is determined by the crystal's optical axis. We designate the beam that loses energy as the "loss beam", and the one that gains energy, the "gain beam". For a given purpose, sometimes the gain beam is the pump, sometimes it is the signal. Typically it is easier to follow the pump and signal paths, though the gain and loss never change.
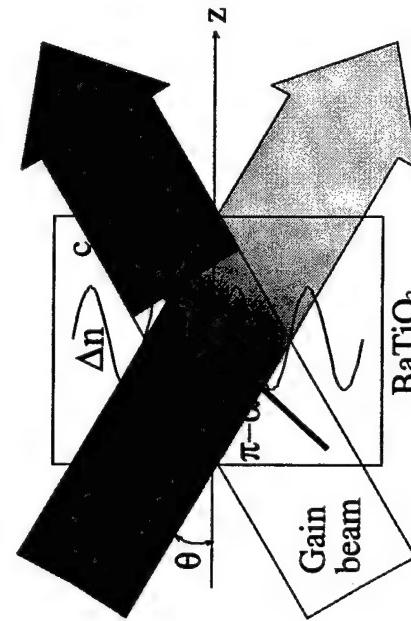
Figure 1.1. Two-beam coupling diagram. Loss beam transfers energy to gain beam. The direction of energy transfer is determined by the c-axis direction and the sign of the charge carriers. Higher gray level saturation implies brightness of the beams.
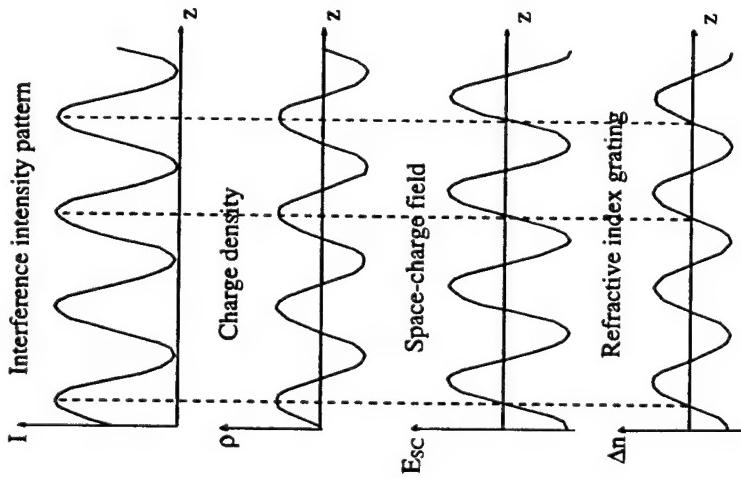
$$\begin{cases} \dfrac{dA_1}{dz} = G A_2 \\[4pt] \dfrac{dA_2}{dz} = -G^* A_1 \end{cases}$$ [1.1]

where G represents the grating strength:

$$G = \frac{1}{2I_0}\gamma\, A_1 A_2^*,$$ [1.2]

and $I_0 = |A_1|^2 + |A_2|^2$ is the total intensity. The gain coefficient γ depends upon the material properties of the crystal and upon the geometrical arrangement of the setup, and is given by:

$$\gamma = \frac{2\pi}{n\lambda\cos\theta}\,r_{eff}, \quad \text{where } r_{eff} = -2 n_o^2 n_e^2\, r_{42}\, E_{sc}(\theta,\alpha)\cos\alpha\sin(2\alpha).$$ [1.3]

In the expression above, r_{eff} is the effective electrooptic coefficient, λ is the wavelength in air of the input beams, θ is the half-angle between the two beams, α is the angle between the beams's bisector and the c-axis, and n_o and n_e, the ordinary and extraordinary index of refraction ($n_o \approx 2.437$, $n_e \approx 2.365$). The r_{eff} depends on the polarization state of the fields. In this treatment we only take into account the main component of barium titanate's electrooptic tensor (r_{42}=1640 at low frequencies), since the other components are more than one order of magnitude smaller. For

Figure 1.2. The photorefractive effect. The intensity pattern from two interfering beams create a charge separation with the same periodicity and shape as the intensity variation. The associated space-charge field, which is shifted by 90° when no external fields are applied, induces an index grating in the medium via the linear electrooptic effect (Pockel's effect).

In steady-state, the energy transfer process is represented by the following two-beam coupling differential equations with respect to the propagation direction z:

extraordinary fields, this term is proportional to $\cos\alpha\sin2\alpha$. Additionally, the space-charge field is a function of both θ and α.

$$E_{sc} = \frac{iK\frac{k_BT}{q}}{1+\frac{K^2}{k_D^2}}$$

[1.4]

It depends on θ through the wave vector $K = \frac{4\pi}{\lambda}\sin\theta$, and on α through the Debye screening length λ_D, which in turn depends on the effective dielectric constant

$\langle\epsilon\rangle = \epsilon_{//}\cos^2\alpha + \epsilon_\perp\sin^2\alpha$, where $\epsilon_{//}$ and $\epsilon_\perp$ are the DC dielectric constant for an electric field parallel and perpendicular to the c-axis, respectively. Barium titanate displays a high dielectric anisotropy with $\epsilon_{//}=135$ and $\epsilon_\perp=3600$.

Solving the above equations, we get the following expressions for the intensities of the gain-beam I_1 and the loss-beam I_2:

$$\begin{cases} I_1(z) = I_1(0)\dfrac{1+m}{1+me^{-\gamma z}} \\[2ex] I_2(z) = I_2(0)\dfrac{1+m^{-1}}{1+m^{-1}e^{\gamma z}} \end{cases}$$

[1.5]

where m is the input intensity ratio $I_2(0)/I_1(0)$.

Using the above equations, we plot in figure 1.3 the gain and loss beam intensities as they propagate through the crystal, for different values of the gain coefficient, 5, 10 and 20 cm⁻¹, and a input intensity ratio of 100. As expected, energy transfer occurs earlier for higher gain coefficients.

11

Energy transfer in two-beam coupling
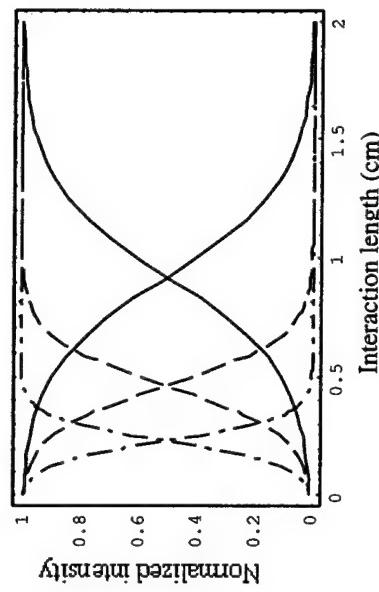
Figure 1.3. Normalized intensity plots of gain and loss beam pairs, $I_1(z)$ and $I_s(z)$ for γ=5 cm⁻¹ (solid curve), 10 cm⁻¹ (dashed curve), and 20 cm⁻¹ (dash-dot curve). The input intensity ratio is set at 100.

Besides the gain, two other parameters are important for evaluating coupling process. The first one is losses, which can be divided into passive losses and active losses. Passive losses are due to surface reflections and absorption in the crystal. Absorption is not taken into account in the above expressions. To account for it, one simply multiplies equations [1.5] by $e^{-\alpha z}$, where α is the absorption coefficient. The curves in figure 1.3 would then be weighed by an exponential decay as they propagate in z. The active losses are due to scattered light from impurities and defects from the crystal, which are amplified by the photorefractive gain, forming what is called "fanning". A picture of fanning simulation is shown in figure 1.4, after the work from [Zozulya, '95]. The losses due to fanning may be reduced by aligning the crystal in the direction of the beam fan, but the trade-off is reduced output signal dynamic range.

12

for the signals in the photorefractive processors is given by Bragg matching considerations and is a few gigahertz for a 1 cm grating. This limit is inversely proportional to the grating length.
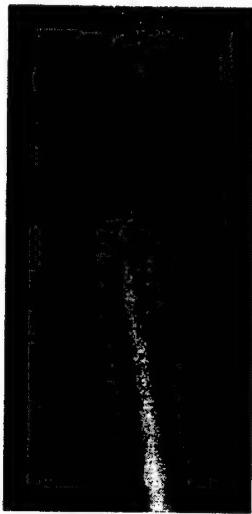
Figure 1.4. Spatial structure of fanning light inside a crystal.

The gratings in a photorefractive crystal adapts to the average input signals integrated over the time constant (τ) of the crystal. The inverse time constant of barium titanate is typically between 1-100 Hz. One might question whether such a slow response doesn't significantly limit the applicability of the photorefractive processors. In some applications, such as phase-conjugation, reconfigurable interconnections and in some image processing systems, a fast photorefractive response is desirable. However, for our applications the slow response time is not a limiting factor, but in fact actually advantageous as it allows signals which are merely 1/τ apart in frequency to be distinguished by the system. In some of these applications, the slow time constant results in long term statistical correlations which allow the system to distinguish even between signals with overlapping bandwidths. For instance, the signals could consist of speech. The speech from a particular person may vary on a short-time scale, however, on a longer time scale a person's speech is more correlated with itself than with another's, which is the key property that allows the speakers to be separated [Min, '88]. On the other end, the upper frequency limit

13

14

CHAPTER 2:

TWO-BEAM-COUPLING MODULES

2.1. INTRODUCTION

The development of photorefractive processing systems, consisting of two or more photorefractive elements, is stymied by the lack of a well-developed component technology which would allow for rapid proof-of-concepts and testing of systems without the burden of extensive, time-consuming element-by-element alignment, as is the case of conventional photorefractive systems.

As a first step towards the development of photorefractive component technology, we developed modular photorefractive units that could be easily interconnected by the use of multimode, optical fibers. The main idea is to have the photorefractive crystal within a module and its associated fiber optic elements optimized for a specific photorefractive function and then permanently fixed in place. The modules are connectorized, and may be interconnected with multimode fibers in any desired configuration. Thus, the assembly of a complete optical system never entails alignment of beams through crystals or other optical elements which are part of each photorefractive function. This means that, with a collection of appropriate modules in hand, a complex system can be quickly assembled or reconfigured.

The geometry of each module can be optimized to perform any of the common wave-mixing effects that have found their way into conceptual and real processing systems, such as two-beam coupling gain [Khuktarev,'79], novelty filtering [Anderson,'89;Horowitz,'91a], four-wave mixing [Cronin-Golomb,'84;Feinberg,'83], self- and mutually-pumped phase conjugation [Feinberg,'82;Sharp,'94;Yeh,'92], photorefractive oscillation [Anderson,'92;White,'82;Yeh,'85], and more. Because it is employed ubiquitously in the photorefractive systems developed in our laboratory, two-beam coupling is the function of choice to test the modular concept.

To this end we have built several two-beam coupling photorefractive modules, such as the one shown in figure 2.1. This chapter discusses the development of these two-beam coupling modules, starting with design considerations, followed by the alignment procedure and the resulting characterization parameters. In the last section of the chapter we test the performance of the modules when these are interconnected to form a photorefractive system, namely an auto-tuning filter, the functioning of which is explained in that section.

Figure 2.1. Two-beam coupling module. The picture shows a ring oscillator which is obtained by simply using an optical fiber to connect the output signal port (gain port) to the input signal port. Another fiber provides the input pump.

2.2. THE MULTIMODE FIBER OPTICS INTERCONNECTIONS

A basic step in designing a component technology for photorefractive systems is determining the kind of interconnections that should be used. The choice is basically between using free-space optics or fiber optics. Multimode fiber optics was selected because, as mentioned above, the use of connectorized fibers dispenses with the need for alignment between modules, therefore offering enormous benefit of ease of assembly and reconfiguration. In addition, the fiber-interconnected modules derive benefits from the substantially well-developed multimode fiber component technology, which provides us with standardized connectors and a gamut of fiber

terminated optical devices. What are the tradeoffs of such benefits? This brief section clarifies the way in which the use of multimode fibers affects the modules.

One major drawback of multimode fibers as opposed to free-space, Gaussian beams is that two-beam coupling gain is reduced, partly because of the nature of speckled beams [Zel'dovich,'95] and partly because of polarization scrambling by the fiber. The electro-optic coefficients in barium titanate are such that the ordinary component of the incident light polarization experiences a negligible two-beam coupling gain, and in addition, it partially erases the grating associated with the extraordinary component. This causes a reduction of the modulation index and, therefore, the gain. Polarization effects can be mitigated by adding small polarizers to the modules, but in our modules we simply accept the sacrifice in gain for the sake of simplicity.

Multimode fibers were used as opposed to single mode, not only because it is much easier to couple a beam into it as compared with the latter, but most importantly, because a multiplicity of modes are necessary for image processing applications, and other photorefractive signal processing applications as well, such as the autotuning filter described in the end of this chapter. A common misconception regarding the multimode fiber optics interconnections is that its use excludes the modules from image processing applications. This is not generally the case. The multimode optical fiber interconnections permit one to process two-dimensional spatial information. Naturally, two-dimensional data becomes scrambled as it propagates down the fiber, however the information contained in the data is preserved provided there are no severely mode-dependent losses. Many image-processing tasks

and algorithms, such as sorting or pattern recognition, are indifferent to scrambling, i.e., the presence of an unscrambled version of the processed image in the output of a system is not essential to its function. Nevertheless, if the unscrambled output image is desired, one can recover such image by holographically correlating the speckle-field output, the scrambled output image, of a fiber-based system with the unscrambled input images. Even if the application is intolerant of scrambling, the fiber-interconnected modular approach may still allow one to explore processing concepts en route to the design of a specialized optical circuit.

Our graded-index fiber interconnections have a 62.5μm core diameter and 125μm cladding diameter. If its numerical aperture of about 0.26 is filled, the fiber will contain on the order of 16,000 modes if illuminated with a 514 nm laser.

2.3. DESIGNS AND RESULTS

As shown in figure 2.1, a two-beam coupling module consists of a barium titanate crystal located in the center of a monolithic aluminum base, and in between two pairs of collets, one input and one output pair. An interbeam angle of about 15° is theoretically calculated to provide maximum gain for barium titanate, however, given the collet's diameter of 3 mm, the interbeam angle between the two collets in a pair is chosen to be 30° as a trade-off to reduce the module's length to about 50 mm. Each collet lies on an independent flexured platform which can be tilted to compensate for small fabrication errors in the parts. This tilt is adjusted by set screws accessible from

the bottom of the base (see figure 2.2). We have tested several base versions with different designs for the flexures and corresponding tilt adjustments.

A top and side view schematic of a module is given in figure 2.2. Two laser beams enter the unit via optical fibers, are collimated by the lenses in the input collets, and then cross in the crystal where they undergo photorefractive two-beam coupling. After traversing the crystal, the loss and gain beams are collected by lenses in the two output collets and focused into another pair of optical fibers. The input and output fibers are terminated with connectors, and the base, along with the connectorized fibers, are assembled in a box, (as shown in figure 2.1).

This design of the modular units optimizes the photorefractive crystal geometry. Small beam diameters between 100 μm to 820 μm allow for small crystals and an efficient use of crystal volume: the length of the crystals typically varies between 2.5 and 4 mm, the cross-section is about $(1.5 \text{ mm})^2$. We get roughly 8 module crystals from a typical $(5 \text{ mm})^3$ barium titanate piece with the appropriate c-axis direction.

We have tested several modules with different combination of base designs, lenses and crystal cuts. Below we discuss the original design followed by our improved, latest design.

Figure 2.2. Schematic with top and side view of two-beam coupling module. A BaTiO$_3$ crystal is placed at the center of the unit over a small glass mount. Collets lie on platforms whose vertical tilt can be adjusted by two set screws. Dimensions are in millimeters.

2.3.1. The original design

The original modules contain the collet design described below and rectangular crystals.

A cross-section of a collet is shown in figure 2.3. The collets contain a 2mm-focal-length aspheric lens (350150 GelTech) which is glued to the collet with UV curing optical adhesive. A 62.5 μm core diameter fiber is inserted into a ferrule, attached with epoxy, and its end polished flush with the ferrule's face to provide an optical quality surface. The prepared ferrule is then inserted into the collet and a specially designed tool, which moves the collet-lens assembly with respect to the

ferrule, is used to collimate a 514 nm laser beam exiting the lens. The two pieces are then fixed with epoxy. The resulting beam diameter is approximately 820 μm.

The crystals were rectangular with a 20° or 30° cut with respect to the crystal axis. Maximum gain is typically achieved for a 45° cut, however the benefit of such gain maximum is associated with unwanted fanning. [Zozulya, A.,'95] Fanning reduces the dynamic range of the two-beam-coupling module, but the effect can be ameliorated by simply reducing the crystal-cut angle, which reduces the gain. The crystal-cut must be tailored to the application at hand. For the applications in our laboratories, high gain is typically desired so the 30°-cut crystals were more successful. All crystals were approximately 4 x 2 x 1.5 mm³.

Figure 2.3. Cross-section of a collet with an aspheric lens (Geltech, 350150). The zirconia ferrule contains a polished multimode fiber with 62.5 μm core and 125 μm cladding. Distance d is adjusted to collimate the outgoing beam. All unspecified dimensions are in millimeters.

2.3.1.1. The alignment procedure

The alignment procedure, as depicted in figure 2.4, begins by mounting the module base on a rotation stage, and placing the crystal on a glass spacer at the center of the base. Next, each of two collets is attached to a jig. The jig consists of a two-dimensional translation stage with an extension arm with a beryllium-copper spring at the end, to which the collet is attached. The collets are positioned 30° apart at the input side of the base. They are made to completely contact the base by having the jig exert a light pressure upon them. With this arrangement a collet has four adjustment degrees of freedom: longitudinal and transversal translation, rotation about the axis through the center of the base and perpendicular to it, which we call "horizontal rotation", and the "vertical tilt" provided by the flexured base. (The translation and horizontal rotation stage are shown in figure 2.4, whereas the vertical tilt, in figure 2.2).

Optimization of the photorefractive gain is the main goal in aligning the input collets. To do this 514nm Argon-Ion laser light is coupled into the input fibers and the overlap of the two beams inside the crystal is then maximized. This is accomplished by use of the longitudinal and transversal translation, and the vertical tilt. (Horizontal rotation is only needed when aligning the output collets.) Typically the gain is not uniform throughout the crystal's volume. [MacCormack,'96] Therefore, one can search for a higher gain by simply moving the crystal sideways, so that the beam overlap probes different regions inside the crystal, and then select a particularly good position. To avoid problems when coupling into the output collets,

the beams must be parallel to the plane of the base. Once the optimal gain is obtained, the crystal and input collets are fixed to the base with epoxy and the jigs are removed.

The same pair of jigs are now used to position the two output collets to optimize the coupling of each beam into the respective output fiber. Finally, the two input and two output fibers are cleaved and terminated in SC connectors and the unit is enclosed in a 10.16 x 7.62 x 3.25 cm³ box.
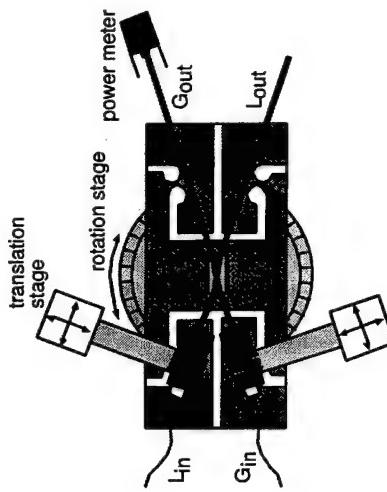
Figure 2.4. Setup for the alignment of the input collets. The gain is optimized by adjusting the translation stages and the set screws shown in figure 2. The rotational stage is used during the alignment of the output collets.

2.3.1.2.Results

Three quantities characterize these modules: gain, passive loss, and time response.

2.3.2. The next generation

This section discusses, based on the above measured results, the routes for improving the module's design towards a new generation. Our primary concern is to reduce the passive losses and, most importantly, to improve the degradation with time of the passive signal coupling due to an alignment drift. Below we list the modifications made in the new design. The alignment procedure is similar to that of the first generation (section 2.3.1.1). The last subsection gives the results for the next-generation modules.

2.3.2.1. A new base design

A top view schematic of the original base is shown in figure 2.5 (a). We call attention to the pivot points of the four flexured platforms. In this original base design, the pivots are wider vertically than horizontally (where horizontal is the plane parallel to the paper in the figure). We'll therefore designate these pivots as "vertical pivots". In some occasions the platforms pivots broke off during the modules' alignment procedure (see section 2.3.1.1 above). This happened because the height adjustment of the flexured platforms provided by these vertical pivots are insufficient for correcting nonidealities of the components in the modules. For instance, the crystal faces are not perfectly parallel, which cause the beams to be deflected. This suggests that the pivots could be significantly mechanically stressed even when they

The gain is measured by taking the ratio of the output intensity of the gain-beam with the loss-beam unblocked over that with the loss-beam blocked. The gain measurements were obtained with an input intensity ratio of the loss-beam over the gain-beam of 3000. Three modules were built with 20°-cut crystals and they had gains of 12, 13 and 19, and another three were built with 30°-cut crystals having gains of 286, 406 and 435.

The passive losses are given by subtracting the passive signal coupling from one. The passive signal coupling was measured by taking the output to input intensity ratio of the gain beam while wiggling the fibers so as to impede the formation of photorefractive gratings in the crystal. The passive losses ranged from 72% to 84%, with an 80% average. We also measured a significant deterioration of the passive signal coupling due to alignment drift with a time scale of weeks. One of the modules passive losses went from 72% to 86% in about a month, whereas two other modules were completely misaligned after the same period.

To test the response time of the modules, we presented a step function to the input and measured the 10 to 90% rise time in the output. The input approximation to a step function, with a rise time of about 5 ms, was generated by manually unblocking the input gain beam. The measurements were performed with a loss beam power of 100 mW and a gain beam intensity of approximately 1% that of the loss beam. The output rise time was always between 500 and 700 ms.

didn't break. And most importantly, such stress is very likely the culprit for the alignment drift as the aluminum tries to relax.

To ameliorate the mechanical stress we tested several different base design modifications. The chosen design is the one shown in figure 2.5 (b), which is similar to the original, but instead of a vertical pivot it has a horizontal pivot (meaning that the pivot is wide in the horizontal plane and thin in the vertical plane). These allow for more range of height adjustment without sacrificing the tilt adjustment.
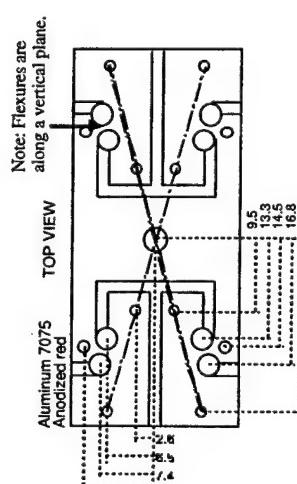
(a)
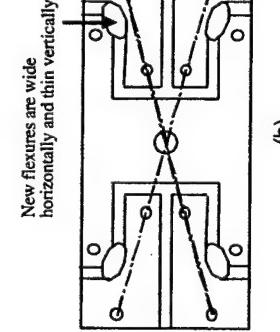
(b)

Figure 2.5. Details of the base design. Dimensions are in millimeters. (a) Original design with vertical pivots. (b) New base design is similar to the original, but with horizontal pivots which allows for greater

adjustment range. The ellipses represent milled slots resulting on wide but thin (about 750 μm) flexures.

2.3.2.2. Brewster-cut crystal

As explained in section 2.2 above, the ordinary (vertical) polarization partially erases the grating, reducing the gain. For two-beam coupling application, the greatest component of the unwanted vertical polarized light typically comes from the pump. So we designed a new "Brewster-cut" crystal geometry as shown in figure 2.6. The Brewster angle for barium titanate is 67°, and is obtained by neglecting the crystal's anisotropy and using an refractive index of 2.4. Having the pump beam enter the crystal at Brewster's angle increases the loss of the vertical polarization component to about 50%, while reducing the desired horizontal polarization component loss to less than 2%. The average reflection loss of the signal at 37° (such that angle of pump minus signal is 30°) is about the same as that at zero degrees, i.e. about 17%. Also, because the pump beam comes in at a steeper angle, it is wider inside the crystal than the signal beam: this facilitates alignment optimization for two-beam coupling. To reduce absorption losses, the new Brewster-cut crystals are 2.5 mm long, as opposed to the previous 4 mm length of the older rectangular crystals. The optical axis of the new crystals are oriented at 45° with respect to the input surfaces.
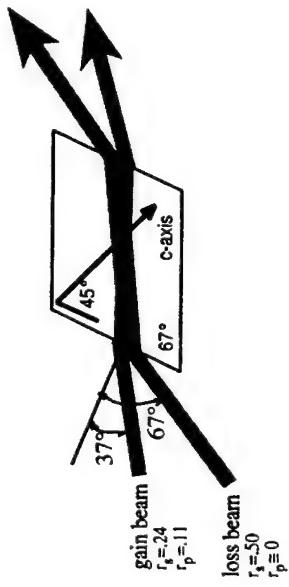
Figure 2.6. Top view of Brewster-cut crystal.

gain beam
r_s=.24
r_p=.11

loss beam
r_s=.50
$r_p \cong 0$

37° 67° 67° c-axis 45°

2.3.2.3. Graded-index lenses

Cheaper than the aspheric lenses, the graded-index (GRIN) lenses come in more variety, are easier to assemble, and dispense the use of collets or other mounts since they can be glued directly to the ferrule using optical adhesive for index matching. Figure 2.7 below shows a cross-section of the new ferrule-lens assembly. It is assembled by using a V-groove mount to align the center of the lens to that of the ferrule while curing the optical adhesive with UV light. The external end of the GRIN lenses were anti-reflection coated to reduce losses. We used a slightly focusing lens pitch (Selfoc SLW1.8 - .29 - 555 from NSG America) to decrease the beam diameter inside the crystal, resulting in beam diameters in the waist of about 100 μm. The time response of a crystal is roughly inversely proportional to the total intensity of the beams. [Horowitz,'91b] This means that the smaller beam diameters should result in a decrease in the response time of the modules.

Graded-Index lens (f=1.8)

V-groove for alignment only

1.8   5.3
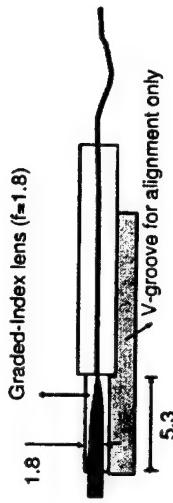
Figure 2.7. Cross section of new lens-ferrule assembly. A schematic of the v-groove is shown but it is removed after alignment. The fiber output will be imaged by the lens at a distance of about 6.3 mm from the output surface of the lens.

2.3.2.4. Buffered fibers

The optical fibers used in the first-generation modules were fibers having only a 245 μm diameter coating over its 125 μm diameter cladding (62.5 μm core). The operation of the modules is sensitive to acoustic vibrations, which disturb the fibers, change the speckle configuration, and thereby affect the two-beam coupling grating. When the modules are tested in a ring configuration such vibrations cause the so called "breathing" in the output, meaning that the output would slowly turn on and off. To ameliorate this effect, we switched to fibers having a 900 μm buffer jacket which helps isolate them from external vibrations. The buffered fiber also has the advantage of greater mechanical resistance, are white, and therefore are easy to see in a darkened room. Not only did we use these buffered fibers in some of our new

modules, but also, and perhaps most importantly, we used them in all the new module-to-module fiber cables.

2.3.2.5. Results for the new generation

We built a total of 3 modules with the above modifications. These modules are characterized in a similar way as described for the earlier modules (see section 2.3.1.2). The resulting small-signal gains are 526, 664 and 843; the passive losses are 68%, 73% and 72%; and the 10 to 90% rise time at the output when the input is unblocked is 120, 164 and 122 msec, respectively.

The higher gain is most likely due to a combination of factors. First, the Brewster-cut crystal geometry, which, as discussed earlier, increases the gain by reducing the ordinary polarization component. Second, the gain characteristics of the particular boule from which the crystals were cut. We have very little control over the material properties of the crystals, making it hard to discern how much each of these two effects contributed to increasing the gain.

The passive-loss average of 71% is about 10% less (better) than that of the first-generation modules. This number is due to a combination of loss components. We measured the absorption coefficient for barium titanate to be about 1.7/cm, resulting in an absorption loss of 35% for a 2.5 mm crystal. The reflection losses are approximately 25%: 50% reflection loss for the ordinary polarization and 0% for the

extraordinary polarization. The remaining 11% losses are consistent with typical measured coupling losses.

The alignment drift was, unfortunately, still present, however it was much less severe than in the first-generation modules. Two months after the above loss measurements were obtained, the measurements were redone and values had changed to 80%, 87%, and 79% respectively, corresponding to an average loss increase of 11%. The loss increase for all three modules is less than the loss increase for the earlier modules.

As expected, due to the smaller beam diameter, these modules responded an order of magnitude faster than the earlier ones. (Even though the pump beam diameter is increased by the Brewster incidence angle to about 250 μm, it is still smaller than the pump beam diameter inside the crystal in the first-generation modules of 850 μm/ cos 15° = 880 μm.)

2.4. DISCUSSION AND FUTURE DESIGN

The second-generation modules provided us with higher gain, lower losses and faster response time than the previous generation. However, an alignment drift is still present, though much less severe than before. This suggests that the components of the modules are performing well, but that the base is still mechanically relaxing causing the alignment drift. We believe the cause of this relaxation is due to the spring action of the flexured platforms. Such relaxation could be enhanced by either

temperature variations or shock. Therefore, as a future project for improving the modules we suggest that a new base be designed. Below we give some suggestions for a new base design, but before that, for the benefit of possible future designers, we first estimate the errors associated with the modules' components.

If further improvement of the base design is pursued, an assessment of errors and tolerances within the module will become necessary. The main misalignment causing errors are the following; the crystal's facet perpendicularity error, typically ± .015 rad, the lens' facet perpendicularity error, ± .006 rad, offset between the center of the lens and the ferrule, and lens tilt with respect to the ferrule. The offset between the center of the lens and the fiber is due to errors in the lens and ferrule radius (± 5 μm for each), lens and ferrule nonconcentricity (negligible compared to diameter error), and the height error in the V-groove mounts used to align the lens to the ferrule (about ± 25 μm). The total offset between the lens and ferrule in the object plane ($\pm\sqrt{(2 \cdot 5^2 + 24^2)} = \pm 25$ μm) is magnified 3.7 times by the lens resulting in a displacement of the beam in the image plane, which occurs inside the crystal, of ± 93 μm. Given the distance of about 6 mm between the lens and the crystal, this displacement is equivalent to an angular error of approximately ± .015 rad. Finally, one edge of the lens can be higher then the opposite edge (by ± 25 μm), which will result in a lens tilt with respect to the ferrule (of ± .004 rad for the 6 mm GRIN lens). This causes an angular deviation of the out-coming beam of ± .006, calculated by using the GRIN lens ray-tracing matrix. Adding up all the contributions we get a total tilt error of $\phi = \sqrt{(.015^2 + .006^2)} = .023$ rad. The current optical length between the

coupling lenses in the module is about 10 mm, resulting in a displacement error of 230 μm, as shown in figure 2.8.
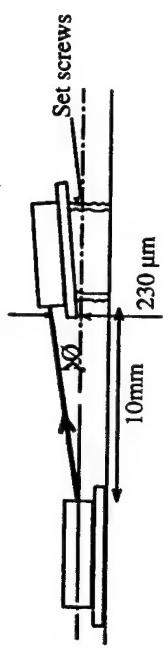
Figure 2.8. Representation of total displacement and tilt errors between lenses.

The main point in designing a new base is the following: Given that no adjustments are needed after the modules are ready, *all the adjustments should be external to the base design*. The base should serve solely to support the components, avoiding the use of internal adjustments (screws, springs, etc.), and thereby reducing the possibility of misalignment after the module is concluded. We suggest a design for the base where the crystal lies in the center of the base, shown in figure 2.9 (a). The five degrees of freedom for each collet would then be externally supplied by a "jig", a 2-D tilt stage which is held by a 3-D translation stage. The main idea is to have the base with poles or blocks that slide up to almost meet the ferrules after they have been aligned. (A design with wide blocks is preferable to one with thin poles, as it is less susceptible to vibrations.) Low shrinkage epoxy is then used to fill the small gap between the ferrules and the base by capillary action. The ferrules should not move during the capillary process as they are held fixed by the jig. The epoxy used to

manipulate the gain/loss trade-off is via the crystal's length, as more length implies more gain but also more absorption losses.

Figure 2.9. Suggestion for modular base design. Ferrule-lens assembly is adjusted externally by translational and rotational stages. Then the corresponding bases are slid upwards to meet the ferrule and glued with low shrinkage epoxy.

Finally, in a stretch of our imagination, it would be interesting if one could find a transparent liquid/gel substance which could solidify with extremely low shrinkage and without changing its refractive index. The ferrules could be positioned, with the use of pre-solidified bars of the same material, inside a mold containing the liquid material. The ferrules would then be aligned and the material in the mold solidified. Cut off the solidified bars that were originally holding the ferrules (which should now be sticking out), polish the six faces (assuming your mold was a parallelepiped) and you have it: a two-beam coupling module as a solid transparent piece (like the fossils in resin that you saw as a kid) with fibers sticking out.

fix the ferrule to the sliding blocks and the sliding block to the base should have low shrinkage coefficient and low coefficient of linear thermal expansion (CTE). It should also have a low enough viscosity (< 1000 cps) so that it can wick into the gap between the ferrule and the pole or block. One viable epoxy is EPOTHIN from Bhueler with a shrinkage coefficient of .0002, CTE of $62 \cdot 10^{-6}/°C$, and viscosity at 25°C of 200 cps. Another possibility is the EP30 from MasterBond with a shrinkage of .0003, which is slightly higher than that of the EPOTHIN but the CTE is lower, $35 \cdot 10^{-6}/°C$, approaching that of aluminum, $24 \cdot 10^{-6}/°C$, and of zirconia, $10.5 \cdot 10^{-6}/°C$, the material used in the ferrules. One could also experiment machining the base out of a material with lower thermal expansion than the currently used aluminum. There are machinable ceramics whose CTE is close to that of the zirconia, such as AREMCO 502-400 with a CTE of $10.8 \cdot 10^{-6}/°C$. The glass-ceramic material Zerodur™ is another possibility with CTE on the order of $10^{-7}/°C$.

Besides a new base design, below are other possible modifications to take into consideration for a future design. Most of them involve a trade-off between gain and loss. One possible modification is to use rectangular cut crystals but with anti-reflection coatings on both input and output surfaces to reduce the reflection losses. However, that would allow both polarizations to enter the crystal and, as discussed earlier, would reduce the gain. That would also significantly increase the cost of the module. On the other hand, one could add polarizers to increase the gain. However, due to the speckle nature of the beams which scrambles the polarization, the polarizers would also increase the passive losses. Naturally, one more way to

2.5. AN APPLICATION EXAMPLE: THE AUTO-TUNING FILTER

Once the modules are in hand, building photorefractive circuits is trivial. The two-beam coupling modules can be quickly and easily interconnected to build and oscillator rings [Yeh,'85], reflexive-coupling units [Anderson,'95], and other systems. We tested the modules by using a ring oscillator with and without a reflexive coupling unit. This architecture is called the auto-tuning filter, an information processing system also known as the feature extractor [Zozulya, A. A.,'95], which we describe in this section as an example of the versatility provided by the modules.

Given two or more temporally and spatially orthogonal signals superimposed on a common input beam, an auto-tuning filter is a self-organizing system that learns to select the strongest input signal from within all the other input signals, in other words, it performs a principal component extraction. It consists of a photorefractive ring resonator with a common input beam as the pump. The non-linear dynamics in the crystal causes a competition of the signals for the resonator modes; the signal with the greater optical energy wins. For simplicity, only two input signals are used in our experiment. We built the auto-tuning filter system in the two different configurations shown in figure 2.10: as a simple ring oscillator and as a ring oscillator with a reflexive-coupling unit inside the ring.

Inputs S1,S2   Output
S1>S2   S1

Inputs S1, S2   Output
S1>S2   S1

(a)   (b)
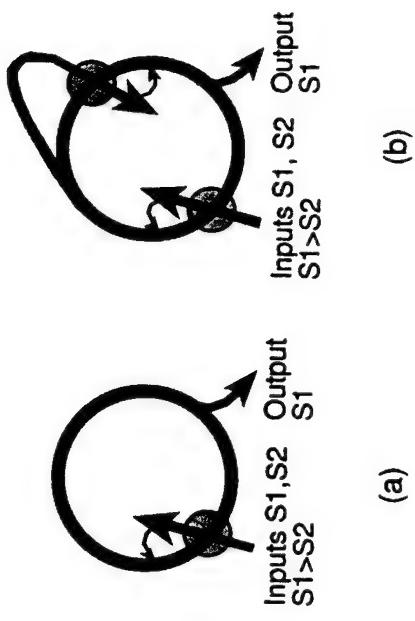
Figure 2.10. Autotuning filter in two different architectures: (a) simple ring, and (b) ring with reflexive coupling. The photorefractive crystals are represented by the circles.

Reflexive coupling [Anderson,'95] increases the competition between the signals that oscillate inside the ring, enhancing the selectivity of the auto-tuning filter. In simplified terms, the reflexive-coupling works as follow: A fixed fraction of these oscillating signals is split away from the ring and folds back to cross the ring inside the reflexive-coupling crystal. Part of these split signals are then coupled back into the ring - the strength of this coupling is dictated by the strength of the photorefractive grating written by each signal. These two signals compete for the available gain in the crystal. The weaker signal writes a weaker grating and thus experiences greater round-trip loss, which results in an even weaker signal, and so on, while the stronger signal undergoes the opposite behavior. Ideally, this positive feedback process continues until the weaker signal is totally inhibited by the stronger

one. In practice though, total inhibition of the weak signal is defeated by fanning [Zozulya, A., '95] of the loss beam into the ring.

Figure 2.11 shows the experimental setup for the auto-tuning filter with reflexive-coupling. The laser beam is split by a polarized beam splitter; the intensity ratio of these two input beams is adjusted by the half-wave plate preceding the beam splitter. Each beam goes through an acousto-optic modulator (AOM) with 60% diffraction efficiency in the first order. The AOMs perform two functions: they shift the optical carrier frequency, resulting in a frequency separation of 280 MHz between the two input beams, which guarantees their temporal orthogonality; and they act as shutters to time modulate the optical carriers with the two input signals, single-sided square pulses at 1 kHz and 625 Hz respectively. The duty cycle of these pulses is set at 90% for low power loss. The two signal beams are then coupled into optical fibers and mixed by a directional coupler. To have the same contribution of each signal in the mixed input, a .50/.50 directional coupler is used (meaning that half of each incoming signal beam is coupled to each outgoing fiber). The beam of one of these outgoing fibers is directed to a photodiode, and the other is photorefractively coupled into the resonating modes of the ring by the pump unit, forming the oscillating beam.

A second directional coupler inside the ring splits the oscillating beam so that 90% is directed into the input loss port of the reflexive-coupling, and the other 10%, into the input gain port. Finally, a third directional coupler (.01/.99) is used to sample the ring, forming the output which is monitored by a second photodiode. The intensity of the the two photodiode signals are displayed on a spectrum analyzer (HP 356A, 100kHz).

The experimental setup for the simple ring oscillator is similar to that of figure 2.11 except that reflexive coupling is bypassed with an optical fiber.
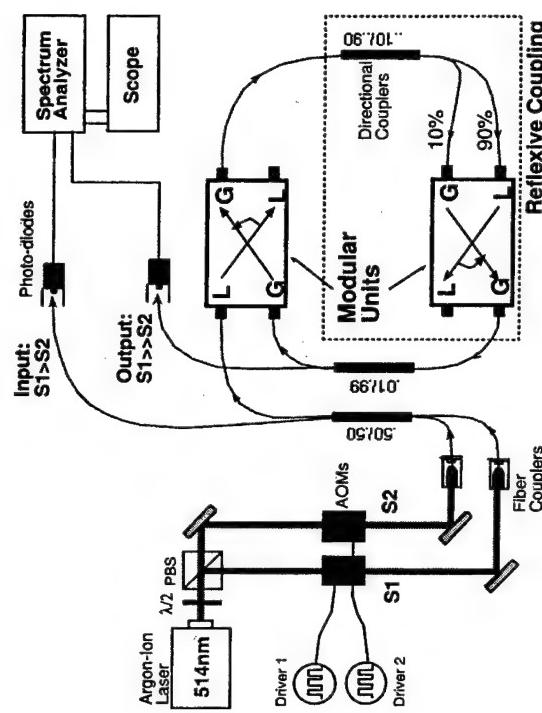
Figure 2.11. Experimental setup for the auto-tuning filter with reflexive coupling. The setup for the simple ring is obtained by simply bypassing the reflexive coupling. (PBS: polarized beam splitter, AOM: acoustooptic modulator.)

The effectiveness of the auto-tuning filter can be given by the contrast ratio (the strongest signal intensity over the weakest signal intensity) in the output, R_{out}, as compared to the contrast ratio in the input, R_{in}. Figure 2.12 shows the results obtained with an R_{in} of 3.17. The intensities of the input signals are shown at the top of the figures and that of the output signals, at the bottom. Figure 2.12 (a) shows the case of the ring with reflexive coupling. The resulting R_{out} is approximately 45 dB. We found

that the system could not self-start oscillation with R_{in} lower than 3.17. This is

beacuse the available gain in the medium is shared by the signals, meaning that the

closer R_{in} is to one, the lower the gain available to the stronger signal. Eventually,

when $R_{in} < 3.17$, this effective gain becomes lower than the minimum necessary for

the system to self-start.

The effect of adding the reflexive coupling to the ring can be evaluated by

comparing the result in figure 2.12 (a) with that in 2.12 (b). The absence of reflexive

coupling in the simple ring configuration decreases the competition between the

signals, resulting in a lower contrast at the output: R_{out} is about 38 dB, 7 dB lower

than with reflection coupling. On the other hand, the removal of reflexive coupling

also reduces the round-trip losses, which enables the system to self-start oscillation

with lower R_{in}. The simple ring could self-start with R_{in} down to 1.75.

Finally, to check for asymmetries in the system, we switched the input signals

of the ring with reflexive coupling, but keeping the ratio constant. The result is shown

in figure 2.12 (c). R_{out} is 43 dB, 2 dB worse than the symmetric counterpart,

corresponding to a 2:3 asymmetry. We believe this asymmetry is mostly due to the

mode structure dependence of the directional couplers.

(a) Ring with Reflexive Coupling:

Input: S1/S2= 3.17 ∓ 0.1
Output: S1/S2= 45.4dB ∓ 1dB   (34 674)

(b) Simple Ring:

Input: S1/S2= 3.17 ∓ 0.1
Output: S1/S2= 38.2dB ∓ 1dB   (6607)

(c) Ring with Reflexive Coupling (symmetric case):

Input: S1/S2= 3.17 ∓ 0.1
Output: S1/S2= 42.5dB ∓ 1dB   (17783)
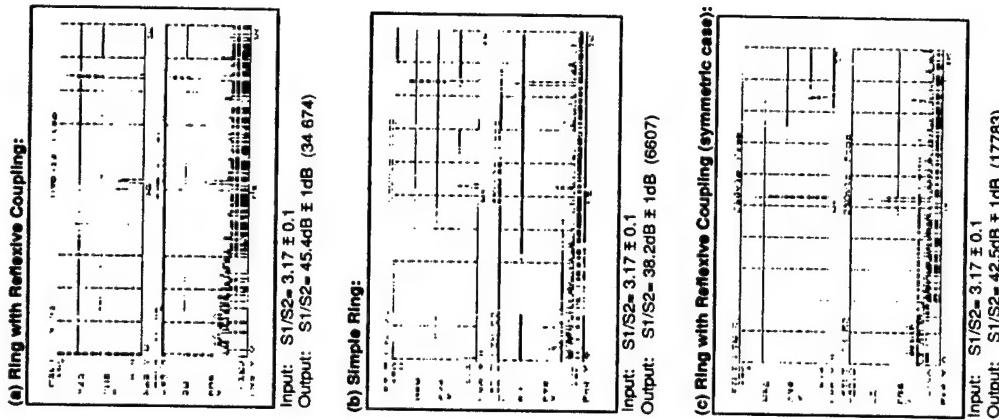
Figure 2.12. Input and output intensity spectra for the autotuning filter. Input spectrum is shown at the top of each screen with a ratio of 3.17:1 between the two input signals (linear scale); output spectrum is shown at the bottom (dB scale). (a) Ring with reflexive coupling, output ratio is 45 dB; (b) simple ring, output ratio is 38 dB; and (c) same as (a), except that the input signals are switched to check for asymmetries, output ratio is 43 dB. Reflexive coupling significantly enhances the selectivity.

CHAPTER 3:

BARIUM TITANATE SPHERES AND SPHERICAL DISKS

3.1.   INTRODUCTION

In the last chapter we discussed the two-beam coupling modules. In this chapter we discuss an idea with the original purpose of making two-beam coupling units even more compact: that of having a spherical barium-titanate crystal as the gain medium. Due to its curvature and high index of refraction, the lensing and gain effect of a photorefractive sphere can be combined to perform fiber-to-fiber two-beam coupling, dispensing the use of external lenses. The resulting two-beam coupling units with barium-titanate spheres are described in section 3.2 below.

We also found that, due to its high photorefractive gain, it is easy to excite whispering gallery modes (WGM) within the spherical crystal by shining a single pump beam onto it. Several patterns were observed, but the lowest order mode, the triangle pattern, was found to be the most robust. The WGM patterns can be better observed and documented when excited in a spherical disk, an equatorial slab of the sphere. In section 3.3 we discuss the oscillating patterns and compare the occurrence region of the triangle patterns with that predicted by a simple theoretical model.

2.6.   CONCLUSIONS

On the auto-tuning filter application above, the use of the two-beam coupling modules allowed quick verification of the properties of the system, such as the fact that adding the reflexive-coupling increases the output contrast. In this context the modules have demonstrably fulfilled its purpose of supplying a means for quick testing of concepts and of new ideas in photorefractive systems. With a collection of different modules available, complex systems can be easily assembled and modified.

The specific modular design one chooses for a particular function, such as two-beam coupling, is always subject to improvements. In this chapter, we have shown the evolution from the original two-beam coupling modular design to an improved design rendering higher gain, lower losses and less long-term alignment drift. Suggestions for a future design to further reduce the drift is also given. Eliminating the long-term drift would add system reproducibility – after a modular system is disassembled it could be re-assembled exactly as before by simply using the same modules.

When building the modules one would want to tailor it to their applications. For example, the number of modules that can be cascaded, say, within a ring, is limited by the losses. Therefore, it might be of interest to further reduce the passive losses of each module, but keeping in mind the typical trade-off between gain and loss, as discussed at the end of section 2.4.

3.2. PREPARATION

Optical quality spheres are straightforward to produce with hand lapping and polishing techniques. [Baxter,'50] We found that barium-titanate and lithium niobate are not substantially more difficult to form into a sphere than optical glass. The barium-titanate was electrically poled in its rectangular form before processing it into a sphere. We feared that barium titanate in particular might become depoled during processing, but such was not the case. Finished spheres of photorefractive barium titanate in fact turn out to be more robust to shocks than their delicate rectangular-cut counterparts, possibly due to the lack of sharp edges or corners from which cracks could propagate.

Spherical disks were produced by first forming a sandwich of two lithium-niobate slabs surrounding a slab of barium-titanate, held together by quartz wax. The sandwich is then lapped and polished into a sphere. Upon removing the lithium-niobate end-caps one is left with a barium-titanate disk with spherical sides, with the c-axis in the plane of the disk. Unlike the sphere, it is easy to observe through the flat top of the disk the beams as they propagate inside the crystal. We produced sphere and disk sizes from 3 mm to 7 mm diameter.

The c-axis of a 0°-cut photorefractive crystal is normally found by placing the crystal between two polarizers and identifying the widely-known conoscopic "iron cross". The lensing property of the sphere morphs the iron cross into a "beach ball" as shown in Figure 3.1, thus it is easy to find and orient the c-axis of the sphere.

Figure 3.1. The "beach ball" pattern of a spherical photorefractive crystal between crossed polarizers observed directly along the c-axis.

3.3. TWO-BEAM COUPLING WITH SPHERES

Two-beam-coupling interaction can be made compact by taking advantage of the geometry of the spheres and the high index of refraction of barium titanate, which is greater than 2 throughout the visible spectrum. Due to these properties, the crystal can directly image the light from one optical fiber to another. In a symmetrical arrangement, the collimated beam inside the sphere can be photorefractively coupled to a second beam propagating between a second pair of fibers. Figure 3.2 shows direct fiber-to-fiber two-beam coupling using a photorefractive sphere without other intervening optical elements. The sphere conveniently magnifies the interaction region viewed by the camera. Two beams, at 514 nm wavelength, cross inside the crystal and undergo two-beam coupling. Each beam is associated with a pair of multimode gradient index fibers, with a 62.5 μm core diameter. With a spherical disk

of 4 mm diameter, the collimated size of the laser beam in the crystal is about 0.8 mm when the fiber-to-fiber imaging is optimized. The apparatus allowed rotation of the crystal axis continuously and permitted the angle between the beams to be chosen from 5° to 45° in 5° increments. Maximum gain occurred at the expected 45° between the c-axis and input beam bisector. The optimum beam angle was observed to be 10°±2.5° - larger than the 4°-6° typically estimated in theoretical plane-wave calculations. [Fainman,'86] Gain was measured using a loss-to-gain beam ratio of 10^5:1 and a gain-beam strength of 0.5 µW. As in the previous chapter, we define gain to be the ratio of the gain-beam intensity at the output of its fiber with the loss-beam present to its intensity with the loss-beam blocked. Because the light from the fibers is unpolarized, one expects an effective reduction of the coupling constant Γ by a factor of 2. We nevertheless obtained a maximum gain of 8000, which is comparable to gains obtained with rectangular-cut crystals of similar length.[1]

Figure 3.2. Two-beam coupling between two pairs of fibers using a spherical photorefractive crystal of barium titanate.

---

[1] The two-beam coupling unit was built and tested by Edeline Fotheringham.

3.4. WHISPERING-GALLERY MODE PATTERNS

We have also investigated interesting oscillation patterns that occur when a Gaussian (free-space) pump is incident on the spherical disk. The oscillating patterns are understood to be low-order whispering gallery modes which are generated by total internal reflection at the crystal's boundaries.

The modes of a spherical resonator that propagate by total-internal reflection at the surface were first analyzed by Lord Rayleigh for acoustic waves and he referred to the analysis as "The problem of the whispering gallery". [Lindsay,'70] Baer demonstrated continuous-wave active excitation of whispering gallery modes using a Nd:YAG sphere pumped with a diode laser. [Baer,'87] Demonstration of active oscillation of whispering gallery modes in a gamut of other materials (and structures) followed, such as in fused-silica [Schiller,'91], polymers [Kuwata-Gonokami,'95], semiconductors [Courtney,'98], and organic materials [Lin,'98]. In these demonstrations oscillation arises from stimulated emission, whereas in our photorefractive spheres oscillation arises from two-beam-coupling, a parametric interaction between the incident pump, the loss beam, and the modes of the resonator, the gain beams.

3.4.1. A family of patterns

At a wavelength of 532 nm, the critical angle for total internal reflection is arcsin(1/2.4), about 25°, thus the lowest order whispering gallery mode of a barium-titanate sphere or disk is a triangle. Since the maximum two-beam coupling gain for barium titanate occurs for small loss-to-gain interbeam angles, one expects the patterns to orient themselves such as to minimize this angle. As shown in figure 3.3 for the triangle and square, the interbeam angle is minimum when a vertex of the pattern aligns itself with either the incident pump location or with the pump's first reflection location. (Second and higher pump reflections are too weak to excite patterns.) We call the former "front-side" patterns (shown with full lines in figure 3.3) and the latter "back-side" patterns (shown with dashed lines). As also shown in figure 3.3, the minimum possible interbeam angle occurs for maximum displacement of the pump, giving 5° for a triangle, 20° for a square and larger for higher-order regular polygons. Since this interbeam angle is smaller for lower order patterns, and smaller interbeam angle translates into higher gain, we expect the triangle to be both the strongest oscillating mode and the easiest to excite.

(a) φ=0°

(b) φ=90°

Figure 3.3. Diagrams of minimum interbeam angles. (a) Pump propagates directly down the center of the crystal. (b) Pump is incident at maximum displacement (r sinφ) from center. In every case, the minimum interbeam angle occurs when vertices are aligned with the pump incidence location (full line triangle and square) or when vertices are aligned with pump's first reflection location (dashed line).

Our experiments reveal that this is indeed the case. The setup used for observing the oscillating patterns is shown in figure 3.4, and pictures of the patterns in figure 3.5. Figure 3.5 (a) shows a front–side oscillating triangle mode in a 5.2 mm diameter spherical disk. In the figure, the pump beam enters the crystal in the vertically downward direction. The pump beam is focused by a microscope objective so that a beam waist occurs inside the disk and is roughly matched in position with a waist of the mode. Most of the pump energy is depleted in the first millimeter after the pump enters the disk, indicating efficient energy transfer. Back-side oscillation

3.4.2. A theory for the triangle occurrence regions

As further investigation we calculate the gain maxima for triangle excitation using a simple model which uses two–beam coupling of the pump with one side of the triangle. The calculations assume plane-waves for the interacting beams. Specifically, we calculate the c-axis orientation which maximizes the gain for the triangle as a function of the incoming pump-beam angle. Figures 3.6 (a) and (b) show the geometrical relations used in our calculations for a front-side and a back-side triangle respectively. For an input displacement of the pump, given by $r\sin(\phi)$, the minimum interbeam angle is given by $2\Theta=30°-\phi'$, where ϕ' is the refraction angle with respect to the surface's normal. The two-beam coupling gain curve, given by the expression (see chapter 1, section 1.2)

$$\gamma = \frac{2\pi}{n\lambda\cos\theta}\, r_{\mathit{eff}}, \quad \text{where } r_{\mathit{eff}} = -2n_o^2 n_e^2\, r_{42}\, E_{sc}(\Theta,\phi)\cos\alpha\,\sin(2\phi), \qquad [3.1]$$

has two maxima when plotted as a function of the c-axis orientation ϕ for a fixed interbeam half-angle θ. Therefore, for each value of the interbeam angle 2θ above we can find from the gain curve two c-axis orientations, ϕ_d and ϕ_r, corresponding to the two maxima. The angles ϕ_d and ϕ_r give the direction of the crystal axis with respect to the bisector of the two interacting beams. The subscript in ϕ_d stands for "direct" denoting that the c-axis direction is less than 90° apart from the bisector, whereas ϕ_r stands for "reverse", and has a value between 90° and 180°. To compare our results to

Figure 3.5. Oscillation patterns inside a barium titanate spherical disk.

also occurs when the crystal c-axis is appropriately rotated. In Figure 3.5 (b) gain is provided to the triangle by the reflection of the pump off of the far-side surface. In this instance the microscope objective is adjusted to collimate the beam in the disk. The oscillation is weaker than the front-side case since the pump intensity is reduced by both absorption and transmission (causing the pump to exit the crystal).

Other modes are excited by changing the orientation of the c-axis, the lateral position of the pump, as well as the pump's angular distribution. Figures 3.5 (c) and (d) show a front-side and back-side square respectively. As explained above, they are much weaker than the triangle because the two-beam coupling gain falls off rapidly with pump-signal angle. We have observed squares in isolation but they were too weak to photograph well. Many patterns can be excited simultaneously under certain conditions. Close inspection of Figure 3.5 (e) reveals a triangle, a square, and a pentagon. For higher angular distribution of the pump, the triangles tend to get blurred, i.e., they also acquire a high angular distribution, as shown in figures 3.5 (c) and (f). Higher order modes are weaker and tend to be difficult to observe both because they experience less gain as the order grows and because they are obscured by the spherical curvature of the disk outer surface. Hexagons were the highest order observed and multiple patterns such as David stars (two interlaced triangles) were also observed.
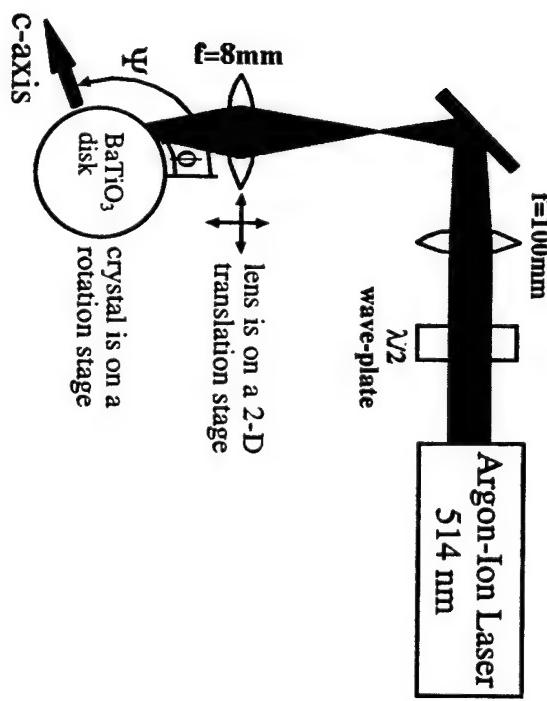
Figure 3.4. Experimental setup for observing oscillating patterns. The angular displacement of the pump is given by ϕ and the rotation of the c-axis by ψNote that the parameters ϕ and ψ are measured with respect to the vertical line (parallel to the pump) which crosses the center of the crystal.

the measurements, we must reference the direct and reverse c-axis angles to the direction of the incoming pump (the vertical in figure 3.6) thereby obtaining ψ_d and ψ_r. Taking both front-side and back-side configurations in figure 3.6 into account, we end up with four optimum c-axis orientations, yielding four theoretical curves of ψ as a function of the input pump-beam angle ϕ.

$$\psi_d = 180° - \Phi_d + \phi + \phi' - \theta \qquad \text{(a)}$$

$$\psi_d = \Phi_d + \phi + \phi' + \theta \qquad \text{(b)}$$
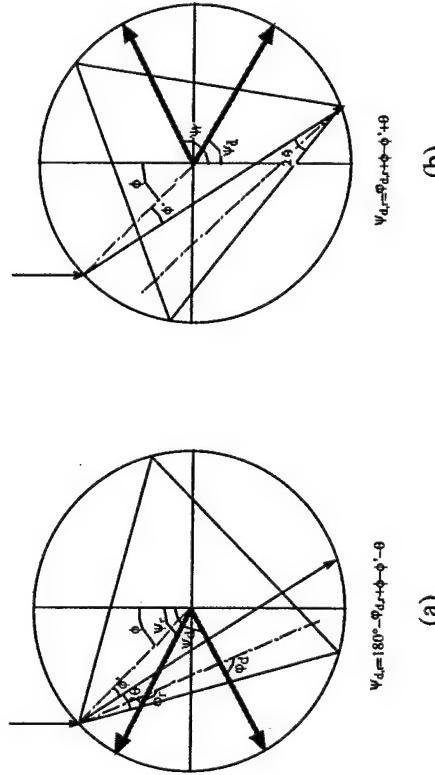
Figure 3.6. Schematic showing the geometrical relationship between ψ and ϕ for a (a) front-side and (b) back-side triangle. For a given θ, there are two optimum values of ϕ that maximizes the two-beam coupling gain, resulting in also two optimum values of ψ, namely ψ_d and ψ_r.

Figure 3.7 shows the occurrence map for triangles with respect to the incident pump-beam angle (ϕ) and the orientation of the crystal axis (ψ). Both angles are

55

referenced to the line that crosses the center of the disk and is parallel to the external incident pump. The four above-mentioned calculated curves (thin lines), corresponding to the theoretical gain maxima, are superimposed over the measured triangle regions (thick horizontal lines). The triangle structures occur over a wide variety of beam parameters. Figure 3.7 shows a reasonable qualitative agreement between the measured regions and the maximum gain calculations. The superposed dotted-lined ellipses are meant to qualitatively indicate the range of parameters over which triangles are observed.
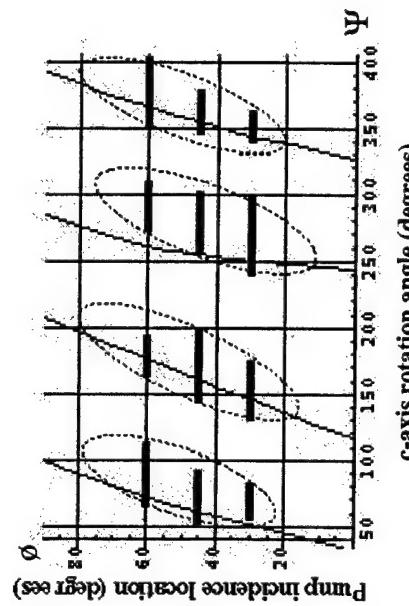
Figure 3.7. Triangle occurrence regions. Dotted ellipses qualitatively show the four regions of the parameter space for which triangles are observed (the pump beam size and divergence are fixed). The horizontal bars show quantitative data regions of triangles. The solid lines represent the four theoretical curves of $\phi(\psi)$ that maximize gain. (The parameter-space coordinates are defined in either figure 3.4 or 3.6.)

56

3.4.3. Unidirectional oscillation

Two qualitative observations suggest that the oscillating patterns are unidirectional and in the forward direction defined by the pump. First, we observe that the light scattered from nonuniformities in the crystal to be in the forward direction. Second, we never observed a steady phase-conjugate beam. In rectangular-cut crystals self-pumped phase-conjugation of a single pump beam is often observed. [Feinberg,'82] The corners of the crystals act as corner-reflectors that generates the phase-conjugated beam via four-wave mixing. Since the disks and spheres have no corners one might wonder whether self-pumped phase-conjugation can occur. In fact, we were never able to observe steady-state phase-conjugation of the input pump-beam under *any* circumstances in any of our sphere or disk samples. We were able to observe highly unsteady phase-conjugation of the pump beam at normal incidence to the crystal, along the c-axis direction.[1]

3.5. CONCLUSIONS

The spherical barium titanate crystal is interesting for several reasons. Most systems built in our laboratories employ two-beam coupling as a basic function. In many cases, the two-beam coupling unit is used to pump a ring oscillator, such as the case of the auto-tuning filter, which we discussed in the previous chapter and will

---

[1] Phase-conjugation observations were done by Dr. Dana Anderson and Dr. Vladimir Shkunov.

revisit in the next chapter. By allowing fiber-to-fiber two-beam coupling, the spheres contribute in building compact systems. Compactness not only has the advantage of making photorefractive systems portable and light weight, but also of reducing vibrations and increasing robustness, this last due to the decreased number of elements necessary in the system. The WGM patterns, for now, are only a scientific curiosity. However, one can think of them as a very compact ring oscillator inside the crystal itself. To make them useful for real systems one needs find a practical way of coupling some of the oscillation light out of the crystal. That could be done by frustrating one of the reflections of the oscillating pattern [Schiller, '92] and then coupling to an optical fiber.

Finally, the spherical disk geometry with the c-axis in the plane of the disk is convenient for determining two-beam coupling parameters. In particular, the internal beam angles are not restricted as they are in rectangular-cut samples by Snell's law and the beam geometry is practically independent of crystal orientation, though there is some degree of optical asphericity due to crystal birefringence.

CHAPTER 4:

CARRIER SUPPRESSION WITH TWO-BEAM COUPLING

4.1. INTRODUCTION

Optical carrier suppression can be used in applications where one wishes to eliminate the carrier or increase the modulation depth of an optically modulated signal. One well known example is in photonic links, where it is necessary that the signal be highly linear and therefore the optical carrier be very weakly modulated. Thus, to avoid the saturation of components upon detection, and to decrease the shot noise due to the information-free carrier, a carrier suppressor might be required to increase the modulation depth.

Several techniques for carrier suppression have been demonstrated. A conventional and straightforward technique is to bias a Mach-Zehnder modulator around the minimum transmission point, such that the optical carrier destructively interferes at the interferometer's output, leaving only the sideband modulations. [Montgomery, '95] Another very common technique for carrier suppression in fiber-optic links is that of using stimulated Brillouin scattering, where the carrier propagating inside an optical fiber pumps a beam which propagates in the opposite direction. Optical feedback of the generated beam enhances the carrier suppression and, in fact, a record carrier suppression of 55 dB has been reported by using a

"Brillouin-erbium fiber laser" configuration. [Loayssa, '00; Tonda-Goldstein, '00] Our goal in this project is to build a simple photorefractive carrier suppressor, which, as opposed to the techniques mentioned above, is adaptive due to the real-time holographic nature of photorefractive gratings. Carrier suppression is achieved by a two-beam coupling interaction within a photorefractive barium titanate crystal and uses the principles of an optical novelty filter [Anderson, '89; Khoury, '91]. Similar techniques have been used for optical heterodyning [Hamel de Monchenault, '88; Khoury, '93], the logical opposite of the carrier suppression problem.

The application of our carrier suppression system is as part of a higher level system, the "optically-smart antenna array". This system, schematically shown in figure 4.1, is composed of an RF-front end active antenna array followed by a photorefractive optical circuit that separates the strongest incoming signal from other weaker ones. For demonstration purposes, the current system uses only two antenna elements and two test-sources, therefore requiring a two-channeled carrier suppressor. The main purpose of the carrier suppressor in this system is to remove the correlation between the modulated optical signals, which is mostly due to the optical carrier, so that the next stage, the autotuning filter, can function optimally. The autotuning filter is the same processor as that described in section 2.5 and its goal, as herein, is to extract the principal component from the various signals presented at its input. Different versions of the auto-tuning filter, some with fiber optics and spherical disks like the ones described in the previous chapter, and others with Gaussian beam optics and other crystal cuts have been tested. However, their use doesn't pertain to the carrier suppressor and will not be considered further here.

This chapter is divided into three further sections. In the next section we give the theoretical background and introduce a geometrical picture to the "operator formalism", an original theory for describing two-beam coupling interactions. This theory proves to be very useful and, in fact, has played a crucial role in the understanding and, thus, in the development of the carrier suppressor. The goal of the next section is not to formally present the complete operator formalism, which will be treated later in the last chapter, but to give a simplified, geometrical view of this theory. It offers an intuitive understanding of two-beam coupling, and thereby, of the carrier suppression process. It provided us the tools necessary to appropriately set the parameters of the system, such as the modulation index or the input beam ratio of the two-beam coupling interaction.

After several iterations of carrier suppressors, and with the help of the operator theory, a system was built which provided a carrier suppression of more than 70 dB, to our knowledge the best published results so far. Section 4.3 describe the experimental setup and results of this system. We compare the results to the theory presented in the section 4.2 by curve-fitting parameters, such as the input intensity ratio, the gain and the input carrier intensity.

Finally, we close the chapter with a discussion of future work and the issues involved in integrating the carrier suppressor to the above mentioned antenna array application. We first discuss how the performance of the carrier suppressor described in section 4.3 fulfills the performance specifications for that system. We then introduce a new carrier suppressor which takes advantage of a channelized electrooptic modulator design, discuss it's implementation, and expose an important

limitation, that of cross-talk between the channels. Thus, further research on the channelized design is still in progress.

Active antenna array   Carrier Suppression   Auto-tuning filter

$S1 > S2,...,Sn$
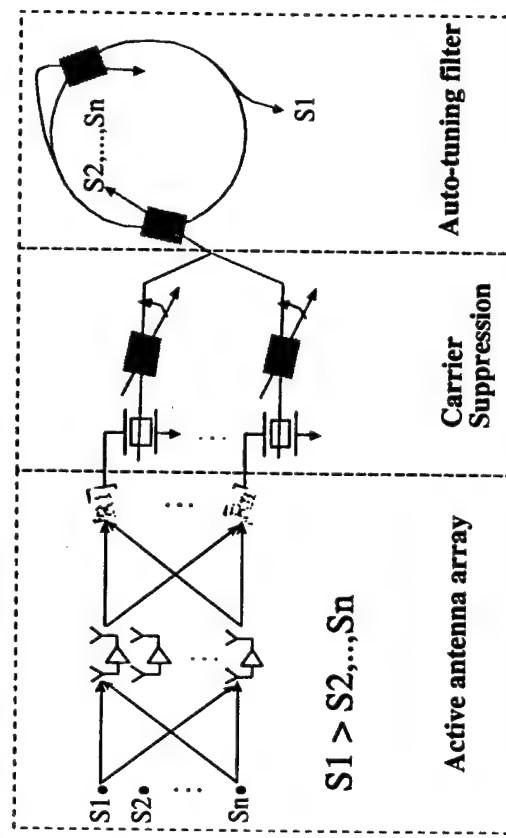
Figure 4.1. Schematic of the optically-smart antenna array, comprising of three parts: the RF front-end which consists of an active antenna array, the carrier suppressor, which consists of electrooptic modulators followed by two-beam coupling in the novelty-filter configuration, and the auto-tuning filter, which consists of a gain crystal pumping a ring with a reflexive-coupling interaction in its feedback path.


4.2. A GEOMETRICAL PICTURE OF TWO-BEAM COUPLING

The current section describes a geometrical picture of the operator theory[1], where the two-beam coupling interaction is viewed as a rotation operator acting on the input fields to produce the output fields. In this simplified exposition of the theory, we assume real coupling, i.e., only pure energy coupling takes place between the beams. With this assumption, the fields can be represented in a two-dimensional space. A geometrical picture of the operator's action over the fields is then constructed. The goal of this section is to provide an intuitive view and understanding of the main aspects of the carrier suppression process. The complete formalism is treated later in chapter 5 and is not necessary in the context of this chapter.

Carrier suppression is schematically shown in figure 4.2. The input to the loss port is the phase-modulated "signal" composed of the optical carrier given by ω_c and the upper and lower sidebands spaced by ω_m from the carrier. The input gain port contains the unmodulated carrier, the "pump". (Note that the definition of signal and pump is opposite than that of previous chapters where the signal beam was the gain port and the pump, the loss port.) In the context of this theory, we refer to the loss port as "minus" port, and to the gain port as "plus". The two beams cross inside the photorefractive crystal where they experience two-beam coupling in the so-called novelty-filter configuration, where any component of the signal that varies faster than

---

[1] This theory was developed by Dr. Dana Anderson.

the response time of the crystal (τ) will not be able to write a grating and will therefore be transferred to the output minus-port, although some of it will passively scatter to the plus-port due to other gratings that may be present. On the other hand, the components which are slower than τ will write a grating and thereby experience active loss by transferring its energy to the gain port. Thus, when experiencing the two-beam coupling interaction the optical carrier in the signal beam will transfer most of its power to the pump beam whereas the sidebands will scatter only some of its power to the pump beam, therefore resulting in the suppression of the carrier at the output signal port. This process assumes that the modulation frequency (ω_m) is greater than the $1/\tau$, typically 1-100 Hz, depending on the crystal sample. The upper bound for the modulation frequency is given by Bragg matching considerations of the modulated beam to the holographic grating written in the crystal. Two frequencies are considered Bragg matched if the difference in phase that they undergo when crossing the crystal's length l is less then π: $\frac{\omega_1 n}{c} - \frac{\omega_2 n}{c} \leq \frac{\pi}{l}$. This results in an upper bound frequency of a few gigahertz for a 1cm crystal. The energy-transfer process depends on the crystal's gain and the intensity ratio between the two input beams. The gain, in turn, depends on the crystal length and other geometrical parameters.

Assuming real coupling, the fieldvector can then be represented in a plane as shown in figure 4.3. Figure 4.3 (a) shows the vectors corresponding to the fields at the input of the system. The carrier frequency is present at both input ports, plus and minus, and therefore is represented by a "carrier" vector E_c, which lies somewhere in the middle of the plane; whereas, since the sidebands are only present in the minus port, they are all aligned with the minus-axis, and can therefore be summed into one single "sideband" vector E_m representing all the sidebands. As we show in the following, a principal eigenvector e can be defined to represent the fields. We also show that the evolution of the system as the fields propagate in the z direction through the crystal, is then represented by a rotation of the eigenvector together with all the corresponding fieldvectors, such that the eigenvector asymptotically approaches the plus-axis. This evolution can thus be represented by a single equation for $\phi(z)$, the angle of the principal eigenvector with respect to the minus-axis. The angle of the carrier vector is labeled $\phi_c(z)$.

The mixing between the fields in the two ports can be conveniently expressed by defining the density matrix

$$\rho = \frac{1}{I} \sum_\omega E_\omega E_\omega^\dagger.$$

[4.1]

where $I = |E_c|^2 + |E_m|^2$ is the total input intensity, ω represents the different frequency components and the dagger represents the vector adjoint. Such representation allows the two-beam coupling interaction to be given by a single equation for $\phi(z)$

[Anderson, 2000 #77]:
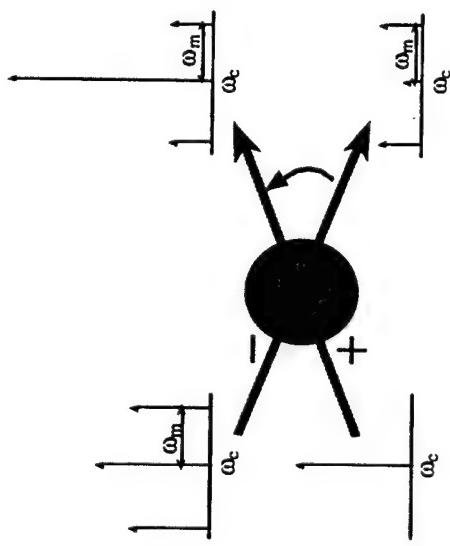
Figure 4.2. Carrier suppression via a two-beam coupling interaction. The input to the minus port is a modulated laser beam while that of the plus port is an unmodulated beam from the same laser. The carrier in the loss port gets transferred to the gain port resulting in a suppressed carrier output.

In the geometrical picture of the operator theory, the different temporal field components of the input signal are represented in a plus-minus coordinate system, defining a "fieldvector"

$$E_\omega = \begin{pmatrix} E_\omega^+ \\ E_\omega^- \end{pmatrix}.$$

To obtain the different components E_ω^+, the input field at the plus port is decomposed into an orthogonal temporal basis, for instance that of harmonic frequencies, and each component or frequency is labeled by the index ω. Similarly for the temporal field components in the minus port, E_ω^-.

65

66

$$2\frac{\partial\phi}{\partial z} = (\lambda_1 - \lambda_2)\sin(2\phi),$$ [4.2]

where λ_1 and λ_2 are the eigenvalues of the density matrix ρ. The normalized interaction length z points towards the propagation direction in the crystal and is defined in units of $2/\Gamma$, where Γ is the photorefractive coupling constant, which in this case is real. In other words, z does not just represent the physical crystal length, but the gain experienced by the fields as they propagate through the crystal.

The solution to the above equation is given by

$$\tan\phi(z) = e^{(\lambda_1-\lambda_2)z}\tan\phi(0),$$ [4.3]

which simply means that the principal eigenvector asymptotically rotates towards the plus-axis as it propagates in z, and that this rotation is "slowed down" by the exponential factor $\lambda_1-\lambda_2$, where $0\leq\lambda_1-\lambda_2\leq1$ since ρ is normalized. It is important to note that, since all the fields in the crystal share the same grating, all the vectors rotate together. One can think of the eigenvector e as "pulling" the other vectors, E_c and E_m, with it.

The main consequence of this rotation to the carrier suppression is the following: As the principal eigenvector rotates towards the plus-axis, it will go through a configuration, shown in figure 4.3 (b), where the projection of the carrier vector on the minus-axis is zero. This means that, for a gain $z=L$ resulting in that particular rotation, there is no carrier present in the output minus-port, i.e., the carrier is perfectly suppressed. It is interesting to note that if the modulation were "off" then $E_m=0$ and the principal eigenvector e would therefore coincide with E_c, such that $\phi=\phi_c$. Thus, E_c would only asymptotically approach the plus-axis, meaning that

67

perfect carrier suppression could not occur for a finite crystal length. However, due to the presence of the modulation E_m, the eigenvector e does not coincide with E_c. Thus, as e approaches the plus-axis, it will go through a particular value of z where E_c coincides with the plus-axis, meaning perfect carrier suppression at a finite crystal length. In other words, *it is the presence of the modulation which allows for perfect carrier suppression at finite lengths.* From the geometry of figure 4.3 (b), it is easy to show that the rotation $\phi(L)$ corresponding to perfect carrier suppression is simply given by

$$\phi(L) = \arctan\left[e^{(\lambda_1-\lambda_2)L}\tan\phi(0)\right] = 90° + \phi(0) - \phi_c(0).$$ [4.4]
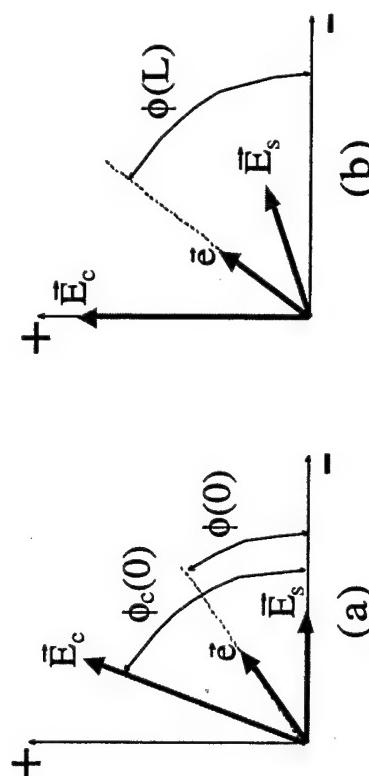
Figure 4.3. Carrier and sideband vectors, E_c and E_s respectively, in the plus-minus coordinate system. The principal eigenvector e embodies the dynamics of the system. (a) Initial position at $z=0$. (b) Position of vectors after rotation, such that the projection of the carrier into the minus-axis is null, resulting in a carrier suppressed output at that port.

Next, we describe the carrier suppression process by using the following experimentally convenient parameters: r is the intensity ratio between the plus and

68

minus input ports; and m is the modulation strength, defined as the ratio between the power in the sidebands over the power in the carrier at the input minus port:

$$r = \frac{\Gamma^-(0)}{\Gamma^+(0)} = \frac{|E_c^+(0)|^2}{|E_c^-(0)|^2 + |E_m^-(0)|^2}; \tag{4.5}$$

$$m(z) = \frac{\Gamma_m^-(z)}{\Gamma_c^-(z)} = \frac{|E_m^-(z)|^2}{|E_c^-(z)|^2}. \tag{4.6}$$

$E_m^-(z)$ and $E_c^-(z)$ are the projections of the fieldvectors $E_m(z)$ and $E_c(z)$ into the minus axis after the corresponding rotation of the input fieldvectors by $\phi(z)-\phi(0)$, where $\phi(z)$ is given by equation [4.3]. Similarly, $E_m^+(z)$ and $E_c^+(z)$ are the projections into the plus axis after rotation. At z=0, the modulation strength is given by $m(0)=m_o$. Using equations [4.3], [4.5] and [4.6], the modulation strength, written in terms of r, m_o, and the gain z, is given by

$$m(z) = \frac{m_0}{\left(1 - \sqrt{r(m_0+1)}\tan[\phi(z)-\phi(0)]\right)^2}. \tag{4.7}$$

We discussed above that, for a fixed value of the input modulation strength m_o, there exists a particular gain (z=L) which results in a rotation such that the carrier is perfectly suppressed, i.e. $m(L)\to\infty$. Conversely, it is also true that, for an arbitrary fixed value of the crystal's gain z, there exists a particular value of m_o for which the carrier is perfectly suppressed. Figure 4.4 shows four different curves for the modulation strength at the output, $m(z)$, versus that at the input, m_o. Each curve has a peak which goes to infinity. The peak occurs at lower input modulation strength for higher gain and lower input intensity ratio. If the input intensity ratio increases from r=1 to r=9, m(z) is higher everywhere except at the peak of the r=1 curve, however,

that has the cost of reduced light intensity in the signal beam (minus beam). Also, if the gain increases from z=6 to z=12, m(z) again becomes higher everywhere except at the peak of the z=6 curve, but now the cost is that the peaks occur for lower m_o.
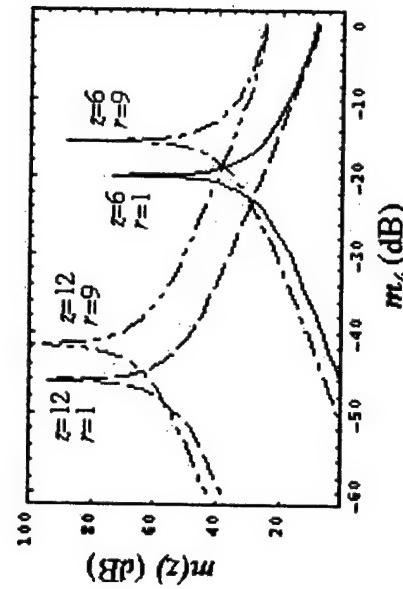
Figure 4.4. Output vs. input modulation strength, for different combinations of gain and input intensity ratio. All peaks go to infinity. Notice that for higher gain the peak occurs for lower input modulation. Except at the peak locations, higher gain or input intensity ratio both result in a general increase of the output modulation, at the cost of decreasing the output signal strength.

In addition, we can find z=L that maximizes m(z) by using equation [4.4], where $\phi(0)$ and $\phi_c(0)$ are written as a function of r and m_o:

$$\phi_c = \arctan\left(\sqrt{r(m_0+1)}\right) \tag{4.8}$$

$$\phi_0 = \arctan\left(\frac{2\sqrt{\frac{m_0}{m_0+1}}}{1-r+\sqrt{(r+1)^2 - 4r\left(\frac{m}{m_0+1}\right)}}\right) \tag{4.9}$$

By plotting L as a function of r, as shown in figure 4.5, we find that, regardless of the input modulation strength, perfect carrier suppression requires the least gain at equal input intensity beams ($r=1$). The above theoretical results are tested in the system discussed in the next section.
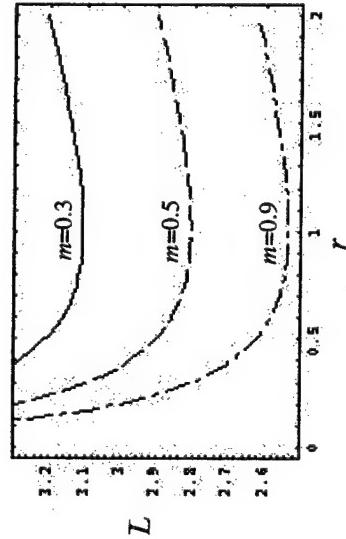

Figure 4.5. Gain required for suppressing the carrier as a function of input intensity ratio for three values of the input modulation strength. Notice that, regardless of the modulation strength, the minimum gain occurs at $r=1$.

4.3. THE CARRIER-SUPPRESSOR

The operator theory introduced in the previous section provided us with the tools necessary to optimize previous generations of carrier suppressors, and build the photorefractive carrier suppressor described in this section.

We prepare the input signals for the carrier suppressor by modulating the minus port optical beam. For that purpose, we built phase modulators containing lithium niobate crystals doped with magnesium oxide (MgO:LiNbO₃). The

71

71

magnesium-doped version was selected as it is known to reduce the sensitivity of lithium niobate to optical damage. [Volk, '94] One drawback of the doping is that it causes striations which vary boule to boule, thereby increasing the chance of wavefront distortion as compared to undoped lithium niobate. However, for our application, wavefront distortion doesn't present any problems as the photorefractive grating can correct for it. A MgO:LiNbO₃ crystal was tested in our laboratory at 532 nm, the wavelength used in the carrier suppression experiment, and resisted developing significant optical damage below 500 W/cm², which for our applications is considered high intensity. Two phase modulators were built, each with a 45.0 x 1.5 x 1.5 mm³ crystal with a half-wave voltage of approximately 70 V at 532 nm. (The wavelength bandwidth for magnesium-doped lithium niobate is from 500 nm to 900 nm.) The optical transmission of 66% through the modulator is rather low – due both to the uncoated optical surfaces, and to striations inside the crystal which are even visible to the naked eye. Each modulator is driven through a resonant circuit mounted directly on the modulator. The resonant circuit, shown in figure 4.6, is tuned to 129.5 MHz and has a quality factor (Q) of about 15. Including the resonant circuit, the modulator has a modulation efficiency of 0.7 rad/V.
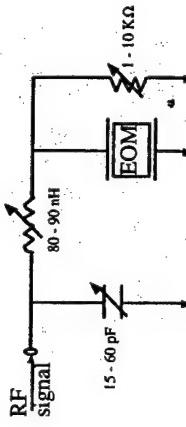
Figure 4.6. Resonant circuit of the electrooptic modulator used in the carrier suppressor.

72

The carrier suppressor is built with 1" optical height standard, resulting in a compact system of about 38 x 38 cm², which reduces the system sensitivity to vibrations and acoustic noise. A picture of the carrier suppressor is shown in figure 4.7 (a). Figure 4.7 (b) shows the setup schematic of the test system. The laser is a doubled Nd:YAG at 532 nm wavelength. A telescope brings the 2.2 mm diameter beam down to about 500 μm. A half-wave plate, followed by a polarizing beam splitter, allows for a variable splitting of the beam. We adjust the splitting intensity ratio so that r=1 at the input of the photorefractive crystal, which, as mentioned in the previous section, minimizes the gain necessary for optimum carrier suppression. One of the beams enters the electrooptic modulator and is modulated by a 129 MHz signal. Since optimum photorefractive interaction occurs with pure-extraordinary input polarization, a polarizer with a 10^{-5} extinction ratio is placed at the output of the modulator to reduce the ordinary polarization component from the signal beam. This beam then crosses the unmodulated beam inside a barium-titanate crystal. A heterodyne detection scheme[1] enabled the direct measurement of the output carrier and modulation sideband strengths. The heterodyne beam is shifted by 80 MHz with respect to the optical carrier via an acoustooptic modulator.
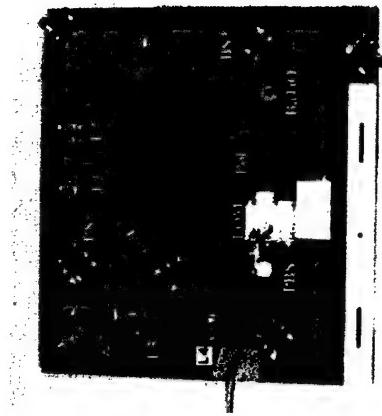
(a)
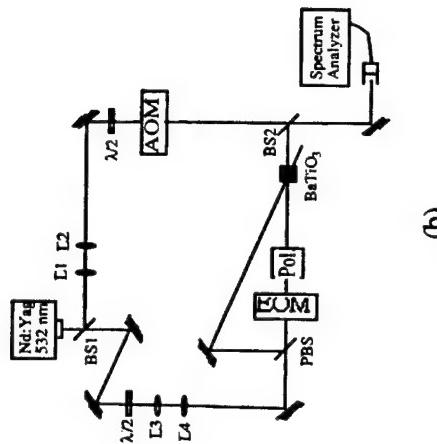
(b)

Figure 4.7. (a)Picture and (b)Setup of the carrier suppressor. (At the time the picture was taken the acoustooptic modulator had been temporarily removed. BS: beam-splitter, PBS: polarizing beam-splitter, L1 and L3: 20 mm focal-length lenses, L2 and L4: 6.15 mm focal-length lenses, EOM: electrooptic modulator, AOM: acoustooptic modulator, λ/2: half-wave plate, Pol.: 10^{-5} extinction-ratio polarizer. All unspecified optics are mirrors.)

[1] Suggested by Bruce Tiemann.

To characterize this system we varied the input modulation strength imposed by the modulator and observed the resulting carrier output as displayed by the spectrum analyzer. From a theoretical standpoint, we could have also varied the gain, but this is not as easy to do experimentally in a well-controlled manner. The measured intensity of the carrier at the minus output port (I_c) is plotted in figure 4.8 as a function of the modulation index (β).[1] A theoretical curve is also shown. We notice two dips of the output carrier strength. The dip at β=2.4 corresponds to a carrier nulling due to the phase modulation process. This zero is not of interest as, first, it is not adaptive and tends to suffer drifts, and second, at this modulation index, the modulation of the signal component is highly nonlinear and has negative slope. Our interest is focused in the first dip occurring at a modulation index of β≡0.11, which is due to the photorefractive carrier suppression. The parameters to the theoretical curve are found by least-squares curve-fitting the data to the theory presented in the previous section. Both the theoretical curve and the data are referenced to the value of the output carrier power when the suppressor is "off", i.e., except for passive losses, the output beam is a replica of the input beam. However, since this reference could not be reliably measured, we included it as a parameter in the fit, along with the input intensity ratio r, and the normalized length z. We also assumed a noise floor to avoid numerical instabilities caused by the infinite dips in the fitting function. The noise floor was set at the measured noise floor of the system, −100 dBm as given by the spectrum analyzer. This value was due to a combination of

---

[1] This particular set of data was taken by Amy Sullivan.

75

noise in the detection process and leakage of the carrier through the acoustooptic modulator in the heterodyne detection beam. The parameters r and z were found to be 1.22 and 6.47, with standard errors of 0.16 and 0.04, respectively.

This first dip shows an output carrier suppression of −71.4 dB ± 1.3 dB. The theoretical curve, the data and the noise floor at −76.8 dB is showed with respect to the fitted reference, which is normalized to 0dB. Notice that at the higher data values the fitted function is higher than the data due to systematic errors. Otherwise, there is good qualitative agreement between the theoretical and experimental curves. The average error between the data and fitted function is 3.7 dB.
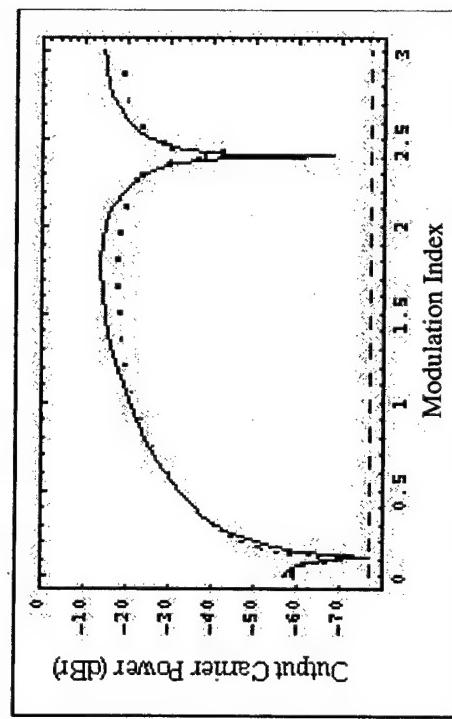
Figure 4.8. Measured output carrier power as a function of modulation index. The output carrier is normalized to the value obtained when the carrier suppression is "off". The first dip of −71.4 dB is due to the photorefractive interaction, whereas the second dip is due to a null of the carrier at the phase modulation process (when β=2.4).

76

From the curve in figure 4.8 we obtain the curve for the output vs. input modulation strength, which is shown in figure 4.9 (compare to the theoretical plots shown in figure 4.4). This curve results in a peak output modulation strength of about $m(z)$=44 dB.


Figure 4.9. Output vs. input modulation strength obtained from fitted carrier suppression data.

As discussed earlier, the original application for the carrier suppressor is as part of the optically-smart antenna array system. The next section discusses how this carrier suppression fulfills, on a per-channel basis, the *performance*-requirements for the antenna array system. However, the antenna array system has *space*-requirements, demanding a re-design of the carrier suppressor to process several channels in a compact space. In that section, we introduce a design that can potentially fulfill these requirements, and discuss its current limitations.

4.4. FUTURE WORK

As described in the introduction to this chapter, the application for our carrier suppressor is as an intermediate stage of the optically-smart antenna array (see figure 4.1), where its function is to provide uncorrelated signals to the following stage, the autotuning filter, by suppressing the carrier. The goal of the work described in this chapter is to demonstrate carrier suppression performance necessary for integration with the antenna array.

A carrier suppression of more than 70 dB was obtained which is above the 30 dB estimated to be necessary at the input of the autotuning filter. Additionally, the signal-to-pump ratio of 1:1 makes the minimum input power for the autotuning filter of 10 mW easily attainable.

After integrating the carrier suppressor with the autotuning filter, the next step is the integration of the optical processor, consisting of both the suppressor and the filter, to the RF front-end. In order to maintain optimum carrier suppression, for a particular gain of the photorefractive crystal, it is necessary that the input modulation strength m_0 be maintained at a specific value which results in a peak of the modulation depth, as discussed in section 4.2. For a variable input RF signal, it becomes necessary to have a mechanism that automatically adjusts the modulation strength to the optimum level. We suggest the use of an automatic gain control amplifier in front of the modulator's driver.

Adequate carrier suppression performance is demonstrated in this chapter, however the incorporation of the carrier suppressor in the antenna array system requires a design that can process several channels in a compact space. The antenna array system's first iteration contains only two channels, but the design should be expandable to more channels for future iterations. In the following, we introduce a double-channeled carrier suppressor design that fulfills these requirements. Such design is still under development as the first implementation attempt ran against some cross-talk problems, as discussed below.

The main difference between this new carrier suppressor and the previous one resides in a new electrooptic modulator design, shown in figure 4.10. This modulator design integrates two modulation channels in the same crystal. The modulator's input laser beam is shaped into a vertical line and the upper and lower portions of the beam are modulated by different signals. The modulated beam then crosses a similarly shaped but unmodulated beam inside the photorefractive crystal, suppressing the carrier from the output beam and thereby revealing the modulation structure of the beam. The channelized modulator design can be expanded to integrate more than two channels without significantly changing the size of the optical system. In fact, using this compact modulator design, we built a 41 x 20 cm² double-channel carrier suppressor, more compact than the previous single-channel one, even though they both utilize 1"-optics. Figure 4.11 shows a picture of this system with its corresponding setup.
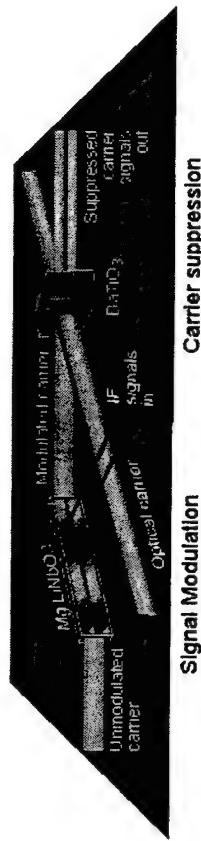
Figure 4.10. Channelized electrooptic modulation and carrier suppression. The modulator imposes spatial and temporal modulation on the optical carrier. The optical carrier is then suppressed by a photorefractive two-beam coupling interaction, revealing the spatial modulation.

The channelized electrooptic modulator contains a thin 0.7 x 5.2 x 45.5 mm³ slab of magnesium-doped lithium niobate crystal, where the optical facets have been cut at Brewster's angle, about 65°, in order to minimize reflections. The electrodes consist of two copper strips, approximately 2 mm wide and separated by about 1.5 mm. Each electrode is separately fed with a narrow-band matching circuit similar to that used in the modulators in the previous section. The center frequency is set to 105 MHz giving a modulation efficiency of about $\eta=0.4$ rad/V.[1]

Other than the channelized electrooptic modulator, there are very few differences between the setup of this system and that of the previous one. To couple light into the electrooptic modulator, a 30° laser line generator lens (from Edmund Scientific Inc.) is used. The heterodyne beam for the output measurement is split directly out of the pump beam. The transmission through the modulator in this setup was about 85%.

---

[1] Dr. Stefania Romisch collaborated with the modulator's assembly and characterization.

However, when testing this system, we found a severe limitation - that of cross-talk between the two channels. The cross-talk was first revealed by a qualitative, visual method: by using the photorefractive crystal to suppress the carrier, we drive each channel, one at a time, and visually check how the signal beams overlap at the output. We found a significant overlap. To quantify this overlap, we measured the beam directly at the output of the modulator. The data in figure 4.12 shows the intensity of the signal component, the carrier, and second order sideband, as seen by the spectrum analyzer. The measurements were made by moving the photodiode detector along the beam line and optimizing the alignment of the heterodyne detection at each point. The modulator was driven at 105 MHz with 25 dBm. In channel 2 the modulation index achieved higher values (going past the dip at $\beta=2.4$) than that in channel 1. We find a significant cross-talk between the two channels with the signal from channel 1 extending significantly over the other channel's side, resulting in a cross-talk of about –10 dB. By changing the coupling alignment into the modulator crystal, we could vary which channel displayed the worst coupling into the other channel's spatial domain, and even adjust for a symmetric coupling, however, the cross-talk remains.

(a)
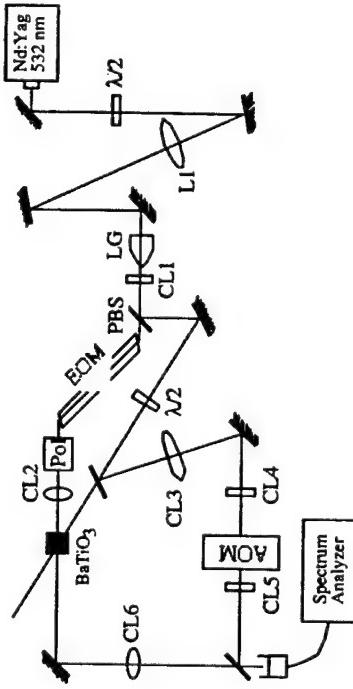
(b)

Figure 4.11. (a) Picture and (b) corresponding setup of carrier suppressor with the channelized electrooptic modulator. (Component abbreviations are the same as those in figure 4.9. Additionally, LG: 30° line generator, CL: cylindrical lenses (CL1: $f_y=8$ mm, CL2: $f_x=30$ mm)).

modulator. To attack the cross-talk issue, research in this new system design is still

under progress.

Alternatively, one could develop a new design which can potentially process

several channels. For instance, we suggest using a grating to generate separate beams

which can be made parallel by an adjustable prism system. The beams would then

enter separate modulators, which however are mounted on a single block. RF

coupling could be reduced by a metal shielding between the driving circuits, while

optical coupling would be eliminated by the air space between the modulating

crystals.

Intensity of modulation components in channel1 vs. position on line

(a)

Intensity of modulation components in channel2 vs. position on line
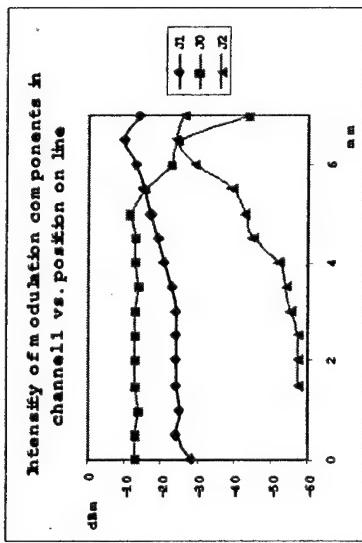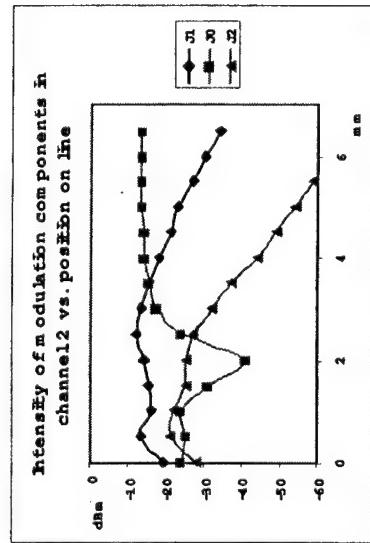
(b)

Figure 4.12. Output intensity of the carrier (J_0), signal (J_1) and second order component (J_2) on (a) channel 1 and (b) channel 2. The signal on channel 1 is more weakly modulated and extends into the spatial domain of channel 2.

The origin of the cross-talk is currently a matter of debate. The most likely

cause is the RF coupling between the resonant circuits that drive the electrooptic

CHAPTER 5:

AN OPERATOR APPROACH TO TWO-BEAM COUPLING:

THE 2-BY-2 COMPLEX-COUPLING CASE

5.1. INTRODUCTION

In this chapter we present an operator theory for describing the process of two-beam coupling. This photorefractive interaction is represented by a black box which operates over the input optical fields to produce the output fields. The input and output fields are represented by vectors, the "fieldvectors", and the black box transformation by the coupling operator T:

$$E_{out} = T E_{in} .$$ [5.1]

The operator theory is based on the same underlying physics as the standard approach for two-beam coupling. [Kukhtarev, '79] However, the nature of operators confers to this theory its main advantage over the standard approach: a mathematical notation that can deal with complicated spatial modes. This lends the power of representing the information as is the most convenient for the information processing application in hand. For example, if the input optical signals are faces, in the standard approach the faces are decomposed into a superposition of plane-waves, and a

differential equation would result for each pair of plane waves, resulting in a complicated problem. With the operator theory the spatial basis may be chosen to be a set of orthogonal facial features, bridging the gap between the mathematical notation and the information processing task at hand.

Because the operator theory is based on the same physical model as the standard approach, it shares the same limitations and approximations. In particular, this theory assumes steady-state gratings, in other words, the spatial and temporal variations of the fields must be such that the gratings reach a stationary state. This means that the temporal components should vary fast enough such that a component in one beam only writes a grating with the corresponding component in the other beam, and with no other. At the other end, the temporal variations should not vary as fast as to cause the Bragg degeneracy between different optical frequencies to be broken. As mentioned earlier in this thesis, for barium titanate, these conditions limit the frequency components to lie between about hundreds of hertz and a few gigahertz for a 1cm crystal. This theory also assumes that the different spatial components are Bragg distinct, in which case we can think of each spatial component as having its own independent grating. Finally, the operator theory assumes that the coupling between the beams in each port are all described by the same coupling constant Γ. This coupling constant is in general complex, meaning that both energy or phase exchange may occur between the coupled beams. [Yeh, '93]

$$\Gamma = |\Gamma| e^{i\gamma} .$$ [5.2]

The fieldvectors in equation [5.1] has two components, E_+ and E_-, corresponding respectively to the gain and loss port of the two-beam coupling interaction. In their turn E_+ and E_- can be decomposed into their various spatial modes. In general, the number of orthogonal spatial modes at the gain port, p, may be different of that in the loss port, m.

$$E = \begin{pmatrix} E_+ \\ E_- \end{pmatrix} = \begin{pmatrix} E_{+,1} \\ \vdots \\ E_{+,p} \\ E_{-,1} \\ \vdots \\ E_{-,m} \end{pmatrix}$$

[5.3]

The operator theory for the general case of $p+m$ spatial modes can be solved numerically and are treated in [Anderson, '00; Anderson, '99], and its details will not be covered in this chapter. Our interest is in the simple case of $p=m=1$, meaning that there is a single spatial component in each port, resulting in a 2x2 dimensionality for the coupling matrix T. For this "2-by-2 case", there exists closed-form solutions for the two-beam coupling evolution and the coupling matrix T.

This chapter follows with three sections. Section 5.2 introduces the general operator formalism.[1] Section 5.3 treats the 2-by-2 case with complex coupling culminating with the solution of T for this case. We also present the special cases of purely real and purely imaginary coupling.

[1] The general theory was developed by Dr. Dana Anderson.

Back in chapter 4, section 4.4, we presented a simplified geometrical picture of the operator approach for the 2-by-2 case, assuming purely real coupling and real input fields. Although the current chapter stands alone, the simplified picture presented in chapter 4 may help lend a preliminary intuitive understanding of the operator algebra. Other approaches using an operator or matrix formulation for two-beam coupling are given in[Liu, '93; Ringhofer, '00; Stojkov, '92].

5.2. THE OPERATOR FORMULATION

Given spatially and temporally varying beams as the input to the two-beam coupling, the fields can be decomposed into a spatial basis, as shown in equation [5.3], as well as into a temporal basis. We will label the different temporal components of the input beams by the subscript ω, but it is understood that these components are not necessarily harmonic components, but any orthogonal temporal basis. The evolution of each temporal component as it propagates through the crystal basis. The evolution of each temporal component as it propagates through the crystal is given by equation [5.1], rewritten here as

$$E_\omega(z) = T(z) E_\omega(0),$$

[5.4]

where z, the interaction length in the direction of propagation, is measured in units of $1/|\Gamma|$, where $|\Gamma|$ is the coupling constant modulus, so the only part of the coupling constant that appears in our equations is it's phase factor γ as defined in equation

[5.2]. The field values at any position z within the crystal is determined by solving for the coupling operator T. As we will see later, T corresponds to a rotation operator which conserves the length of the fieldvectors (meaning that the transformation is lossless) and is therefore Unitary, i.e. $T^\dagger = T^{-1}$. The formulation involved in solving for T follows.

In an analogy to quantum mechanics, we introduce a "Hamiltonian" H, a transformation that determines the evolution of the fieldvector such that

$$E'(z) = -iH(z)E(z),$$ [5.5]

where the prime denotes derivation with respect to z. The Hamiltonian operator represents the effect of the grating on the evolution of the fields in each location z. It will later be defined such as to embody the two-beam coupling physics in the context of our operator formulation, but because the grating itself is formed by the fields, we can anticipate that $H(z)$ will depend on the fieldvectors. However, this fact does not affect the validity of equation [5.5].

The black-box view, given by the coupling operator T, and the above fieldvector evolution, given by H, can be connected by simply substituting equation [5.4] in the above equation, giving

$$T'(z)E_\omega(0) = -iH(z)T(z)E_\omega(0),$$ [5.6]

which, by eliminating $E_\omega(0)$, gives us a differential equation for the coupling operator:

$$T' = -iH\,T,$$ [5.7]

where the explicit z-dependence is dropped.

Our problem involves the mixing between two input beams or ports, which translates into a mixing of the components of the fieldvector. Thus a "density operator", ρ, which represents the mixing between the components is introduced:

$$\rho = \frac{1}{I}\sum_\omega E_\omega \times E_\omega^\dagger,$$ [5.8]

where $E_\omega^\dagger$ is the adjoint (complex-transpose) of E_ω, and I is the total intensity:

$$I = \sum_\omega I_\omega = \sum_\omega E_\omega^\dagger E_\omega .$$

In the 2-by-2 case, the density operator in the gain and loss port space is given by the density matrix:

$$\rho = \frac{1}{I}\sum_\omega \begin{pmatrix} \left|E_{\omega,+}\right|^2 & E_{\omega,+}E_{\omega,-}^* \\ E_{\omega,+}^* E_{\omega,-} & \left|E_{\omega,-}\right|^2 \end{pmatrix}$$ [5.9]

Characterizing the system by the density operator is equivalent to characterizing it by its fieldvectors, with the advantage that it simplifies and suits the problem. [Cohen-Tannoudji, '77b] The density operator is Hermitian, and, because it is normalized, it has a unity trace:

$$Tr(\rho) = \sum_i \rho_{ii} = 1.$$

Just as we had represented the fieldvector in the black-box view of equation [5.4], we can equivalently represent the density matrix at some location z as a function of the input density matrix by simply substituting equation [5.4] in equation [5.8], obtaining

$$\rho = T\rho(0)T^\dagger.$$ [5.10]

And, again, equivalently to equation [5.5] for the fieldvectors, we can write a differential equation for the density operator, which is obtained by combining the above equation [5.10] with equation [5.7], giving

$$\rho' = -iHT\rho(0)T^\dagger + i\,T\rho(0)T^\dagger H = -i[H,\rho],$$ [5.11]

where $[A,B] = AB - BA$ is the "commutator" of A and B.

The equations above hold independently of the physics of the problem. As mentioned above the physics is included by defining an appropriate Hamiltonian, which satisfies the assumptions and approximations of the physical model described in the introduction of this chapter. It is given by

$$H = \frac{i}{4} e^{i\sigma_3}[\sigma_3,\rho],$$ [5.12]

where ρ is the same density operator defined above and σ_3 is an operator which embodies the asymmetry of the coupling mechanism, and, in the 2-by-2 representation, is given by the Pauli spin matrix

$$\sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

The factor of i in the Hamiltonian makes it Hermitian. This two-beam coupling Hamiltonian is not derived here from first principles, but, upon substitution in equation [5.11], the resulting equations for the 2-by-2 case can be shown to reduce to that of the standard approach. [Anderson, '00]

5.3. A CLOSED-FORM SOLUTION TO THE 2-BY-2 CASE.

The operators defined in the previous section are valid in the general case, when a superposition of spatial modes are present in each beam (the n-by-n case). We now focus on the 2-by-2 case where only one spatial mode is present in each beam.

For this simple case, closed-form solutions exist for both the density operator ρ and

the coupling operator T. From this point forward, we will cease to refer to them as

operators but instead refer to their 2-by-2 matrix representation.

Any 2-by-2 matrix can be represented by a superposition of the unit matrix

and the Pauli spin matrices. As we progress on the problem of deriving the coupling

matrix, we will see that such a representation proves to be very well suited for the

density matrix [Cohen-Tannoudji, '77a]:

$$\rho = \frac{1}{2} + \sum_{i=1}^{3} s_i \sigma_i,$$ [5.13]

where

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \ \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \ \text{and} \ \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

and

$$s_i = \frac{1}{2} Tr(\rho \sigma_i).$$ [5.14]

The Pauli spin matrices have several interesting properties. First, they are idempotent,

meaning that $\sigma_i^2 = 1$. Thus $e^{ia\sigma_i} = \cos a + i\sigma_i \sin a$, where a is a scalar. Second, they

are cyclic upon multiplication, i.e., $\sigma_i \sigma_j = i\varepsilon_{ijk}\sigma_k$, where $i \neq j \neq k$ and ε_{ijk} is 1 for an even

permutation of i,j,k and -1 for an odd permutation. Third, they anti-commute,

meaning that $\{\sigma_i, \sigma_j\} = \sigma_i \sigma_j + \sigma_j \sigma_i = 0$ where $i \neq j$.

The coefficients of the spin matrices can be conveniently expressed in

spherical coordinates:

$$s_1 = \Lambda \sin(2\theta)\cos(2\phi)$$
$$s_2 = \Lambda \sin(2\theta)\sin(2\phi)$$
$$s_3 = \Lambda \cos(2\theta)$$ [5.15]

where $\Lambda = \sqrt{s_1^2 + s_2^2 + s_3^2} = (\lambda_1 - \lambda_2)/2$ where λ_1 and λ_2 are the eigenvalues of ρ, with

$\lambda_1 > \lambda_2$. The ρ-matrix in the spherical coordinates is simply given by:

$$\rho = \begin{pmatrix} \frac{1}{2} + \Lambda\cos(2\theta) & \Lambda\sin(2\theta)e^{-i2\phi} \\ \Lambda\sin(2\theta)e^{i2\phi} & \frac{1}{2} - \Lambda\cos(2\theta) \end{pmatrix}.$$ [5.16]

We prefer to write the solution of the density matrix and coupling matrix in

terms of the spin matrices:

$$\sigma_{/\!/} = \sigma_1 \cos(2\phi) + \sigma_2 \sin(2\phi) = \begin{pmatrix} 0 & e^{-i2\phi} \\ e^{i2\phi} & 0 \end{pmatrix}$$

$$\sigma_{\perp} = \sigma_2 \cos(2\phi) - \sigma_1 \sin(2\phi) = i\begin{pmatrix} 0 & -e^{-i2\phi} \\ e^{i2\phi} & 0 \end{pmatrix}$$ [5.17]

As can be deduced from the expressions above, are orthogonal and are obtained by

rotating σ_1 and σ_2 around the σ_3 "axis", such that $\sigma_{//}$ is aligned with the projection of

ρ into the $\sigma_1\sigma_2$-plane, as shown in figure 5.1.
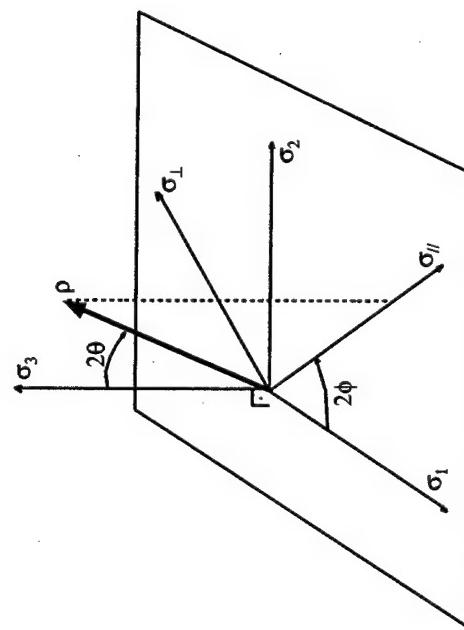
$$\sigma_\rho = \sigma_3 e^{i2\theta\sigma_\perp} = \sigma_{//}\sin(2\theta) + \sigma_3\cos(2\theta),$$

points in the direction of ρ.

The expression for the Hamiltonian is obtained by substituting the above ρ in

equation [5.12], giving:

$$H = -\frac{\Lambda}{2}\sin(2\theta)\big(\sigma_\perp\cos\gamma + \sigma_{//}\sin\gamma\big).\qquad[5.19]$$

Substituting the above H and ρ in equation [5.11], provides the differential equations

for θ and ϕ:

$$\theta' = -\frac{\Lambda}{2}\cos\gamma\sin(2\theta)$$
$$\phi' = \frac{\Lambda}{2}\sin\gamma\cos(2\theta)\qquad[5.20]$$

The solution for θ is straightforward:

$$\theta = \arctan\big(e^{-\Lambda z\cos\gamma}\tan\theta_0\big),\qquad[5.21]$$

where

Figure 5.1. Representation of ρ in σ_1, σ_2, σ_3-coordinate space. The "vectors" $\sigma_{//}$ and $\sigma_\perp$ are the parallel and orthogonal projection of ρ on the $\sigma_1\sigma_2$-plane and they rotate in the plane as ρ rotates in space.

Using these, we can rewrite the density matrix as:

$$\rho = \frac{1}{2} + \Lambda\sigma_\rho\qquad[5.18]$$

where $\theta_0=\theta(0)$ and the same convention will be used from this point forward with all variables. Solving ϕ by using $\phi' = \frac{\partial\phi}{\partial\theta}\theta'$, we get

$$\phi = \frac{\tan\gamma}{2}\ln\left(\frac{\sin(2\theta_0)}{\sin(2\theta)}\right) + \phi_0 \qquad [5.22]$$

where

$$\sin(2\theta) = \frac{\sin(2\theta_0)}{e^{\lambda z\cos\gamma}\cos^2\theta_0 + e^{-\lambda z\cos\gamma}\sin^2\theta_0}$$

$$\cos(2\theta) = \frac{e^{\lambda z\cos\gamma}\cos^2\theta_0 - e^{-\lambda z\cos\gamma}\sin^2\theta_0}{e^{\lambda z\cos\gamma}\cos^2\theta_0 + e^{-\lambda z\cos\gamma}\sin^2\theta_0} \qquad [5.23]$$

The solution for ϕ is more usefully casted as:

$$e^{\pm i2\phi} = \left(\frac{\sin(2\theta_0)}{\sin(2\theta)}\right)^{\pm i\tan\gamma} e^{\pm i2\phi_0} = \left(e^{\lambda z\cos\gamma}\cos^2\theta_0 + e^{-\lambda z\cos\gamma}\sin^2\theta_0\right)^{\pm i\tan\gamma} e^{\pm i2\phi_0} \qquad [5.24]$$

With the solution of θ and ϕ at hand, ρ is now solved. The plot in figure 5.2 shows the shape for the evolution of ρ in the $\sigma_1\sigma_2\sigma_3$-space (as shown in figure 5.1)

for $\gamma=1.5$, i.e., $0<\gamma<\pi/2$. Figure 5.2 (a) shows a view of the south pole, whereas figure

5.2 (b) shows the north pole, which points in the direction of σ_3. Given the initial

conditions θ_0 and ϕ_0, ρ_0 will point to a particular point in the curve and then, as it

propagates through the crystal, ρ will follow the curve asymptotically towards the

north pole. The value of 2θ is an expression of the intensity ratio between the beams.

The north pole corresponds to all the energy being present at the gain port, and the

south pole, to it all being present at the loss port. Thus, naturally, the equator

corresponds to equal intensities between the two beams. On the other hand, 2ϕ

corresponds to the phase-difference between the two beams and the spiraling

indicates a phase coupling between the beams. As indicated by the $\cos(2\theta)$ factor in

equation [5.20] for ϕ, the spiraling in the northern hemisphere is in the opposite

direction as that in the southern hemisphere. At the equator, no spiraling can occur as

the curve is perpendicular to the equator line. This has a simple physical

interpretation. Unlike energy transfer, two-beam coupling phase transfer is

symmetric, and thus, when the beams have equal intensity they both change their

phases at equal rates, resulting in no net change in the phase difference between the

two beams.

the north pole without spiraling. That is indeed the case and is shown in figure 5.3 (a), plotted for $\gamma=0$. Conversely, an imaginary coupling constant, which occurs for $\gamma=\pi/2$, corresponds to pure phase coupling and is represented by a pure rotation around σ_3. We call this the "imaginary case" and it is shown in figure 5.3 (b).

Naturally, the longitude of the curve in the real case is given by the constant $\phi=\phi_0$ and similarly, the latitude of the curve in the imaginary case, by θ_0. In the real case the equation for θ is simply given by substituting $\gamma=0$ in equation [5.21]. In the imaginary case, the complex-case expression for ϕ is not valid as it becomes undetermined at that limit. Notice that when $\gamma \rightarrow \pi/2$ is plugged in equation [5.22], since $\tan(\gamma) \rightarrow \infty$ and $\ln(1) \rightarrow 0$, ϕ is undetermined. However, since in this case $\theta=\theta_0$ is constant, the expression for ϕ directly follows from its differential equation [5.20],

giving $\phi(z) = \frac{\Delta\kappa}{2}\cos(2\theta_0) + \phi_0$.

(a)   (b)
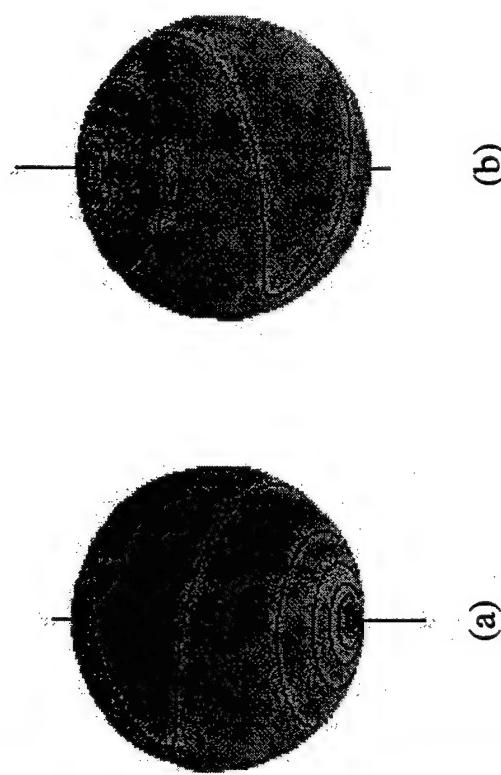
Figure 5.2. Plot of the evolution of the density matrix ρ as a function of propagation z through the medium, for $\gamma=1.5$. The initial conditions are $\theta_0=1.5$, close to the south pole (a), where all the energy is in the loss port, and $\phi_0=0$. As the density matrix evolves, it spirals up and switches direction at the equator, where θ' is maximum $\phi'=0$, and then asymptotically spirals up to the north pole (b).

So, in summary, for the general case of complex coupling, where $0<\gamma<\pi/2$, the energy coupling causes ρ to approach the north pole whereas the phase coupling causes it to rotate around the polar axis σ_3, resulting in the spiral shape shown in figure 5.2. We know from photorefractive physics that a real coupling constant, which occurs for $\gamma=0$ (see equation [5.2]), corresponds to pure energy coupling and thus, for this special "real case" we expect the curve to go directly from the south to

$$\frac{\partial T}{\partial \theta}\theta' + \frac{\partial T}{\partial \phi}\phi' + \frac{\partial T}{\partial z} = i\frac{\Lambda}{2}\sin(2\theta)\left(\sigma_\perp\cos\gamma + \sigma_{\shortparallel}\sin\gamma\right)T \qquad [5.25]$$

which has the following solution for T:

$$T = T_\theta T_\phi T_\rho = e^{-i(\theta-\theta_0)\sigma_\perp}e^{-i(\phi-\phi_0)\sigma_3}e^{i\frac{\Lambda}{2}\sin\gamma\,\sigma_{\rho 0}}.\qquad [5.26]$$

The special limit cases are straightforward by substituting $\phi=\phi_0$ and $\gamma=0$ for the real

case, and $\theta=\theta_0$ and $\gamma=\pi/2$ for the imaginary case:

$$T_{Re} = e^{-i(\theta-\theta_0)\sigma_\perp} \quad \text{and} \quad T_{Im} = e^{-i(\phi-\phi_0)\sigma_3}e^{i\frac{\Lambda}{2}\sigma_{\rho 0}} = e^{-i\frac{\Lambda}{2}\cos(2\theta_0)\sigma_3}e^{i\frac{\Lambda}{2}\sigma_{\rho 0}}.$$

The action of T on ρ is simple: each factor of the type $T = e^{iA\sigma_i}$ corresponds to a

rotation of ρ around σ_i by an amount A. So the complex solution in equation [5.26]

represents a general rotation of the density matrix decomposed into the following

rotations: the first factor to act on ρ, T_ρ, rotates ρ around itself, a degenerate rotation

which does not affect ρ; the second factor T_ϕ then rotates ρ around σ_3 by $\phi-\phi_0$; and

finally the last term T_θ rotates ρ around $\sigma_\perp$ by $\theta-\theta_0$, which is a rotation directly up

towards the north pole (see figure 5.1).

The interpretation of the factors T_ϕ and T_θ are clear, the first represent the

relative phase difference between the beams picked up by the phase transfer, whereas

(a)   (b)

Figure 5.3. Evolution of the density matrix for the special cases of (a) real and (b) imaginary coupling. In the real case of pure energy coupling, ρ follows a longitude line towards the north pole, the longitude being determined by ϕ_0. In the imaginary case, ρ rotates around the pole with constant speed, the latitude being determined by θ_0.

As described above, the solution to the density matrix provides us with an important tool for understanding the behavior of the system, but our ultimate goal is to solve for the coupling matrix which represents the two-beam coupling black-box. If the chain rule is applied to the differential equation for the coupling matrix, equation [5.7], and the expressions for θ', ϕ', and H (equations [5.20] and [5.19] respectively) are substituted, we get:

CONCLUSIONS

In this chapter we discuss applications and future work regarding the different aspects of the research presented in this thesis. This thesis revolves around one fundamental photorefractive interaction, the two-beam coupling effect. The contributions described in this thesis are divided into three aspects of photorefractive systems: The first is that of developing a component technology, which includes the demonstration of the two-beam coupling modules described in chapter 2 and by the study of the oscillating patterns occurring inside a barium-titanate spherical disk, in particular the triangle pattern, described in chapter 3. The second is that of building and optimizing a specific photorefractive application, the carrier suppressor, described in chapter 4. Finally, the third is that of developing an analytical tool that can simplify the modeling and prediction of photorefractive systems. In this case, the contribution is the presentation of a general solution for the operator theory in the case of a single spatial mode in each beam, described in chapter 5. These different aspects sometimes overlap. For example, the operator theory provided a model on which to cast the carrier suppression effect, resulting in an understanding of the system which proved to be crucial for its optimization. Below, we give concluding remarks and related applications on the topics covered in this thesis.

The fiber-interconnected two-beam coupling modules are pre-aligned building blocks for photorefractive system design. They provide a way to quickly test an idea without having to invest the time in critical optical alignment which is necessary

the latter represents the energy coupling. However the interpretation of T_ρ is somewhat subtle. It is interesting to note that if this factor is removed from the solution, the resulting expression still satisfies the relation $\rho = T\rho(0)T^\dagger$ however it will not satisfy the differential equation $T' = -iH\,T$. We believe this is because ρ doesn't have any information about the evolution of the absolute phases of the fields, but only the relative phase between them, so it is indifferent to T_ρ. However, the two-beam coupling equations provide information about how much phase is being transferred from each beam to each other beam, and not only the difference. This information is therefore contained in T by this degenerate-rotation of ρ. For example, we mentioned above that ϕ' is always zero at the equator, however that does not mean that the phase transfer stops at that location, it just means that the phase transfer is symmetrical, and this information is contained in T by T_ρ.

when building a conventional system. However, it is important to note that, because the geometry within a module is fixed, they are not well suited for optimization of systems. The utility of the two-beam coupling modules as a component technology was illustrated in the example given in section 2.5, where the inclusion of a reflexive-coupling unit in the autotuning filter's ring provided an increase of the output contrast between two signals which are being separated. It took only a few minutes to include the reflexive-coupling element and test the performance. Future work on the modules should seek to improve their long term stability. In section 2.4 some suggestions for future designs were given, but the main issue to keep in mind in a new design is that the adjustments be external to the final module, thereby reducing the susceptibility to long term alignment drifts. Among the useful applications of the modules, we suggest that of testing different ring configurations. Because the geometry of the modules are fixed, the differences in behavior can then be attributed to the different architectures and not to the geometrical factors of the two-beam coupling configurations. For example, one can compare the behavior of the autotuning filter architecture shown in figure 2.10 (b) with one where the pump crystal is instead located in the reflexive loop (directly in front of the plus input port of the reflexive coupling crystal). Understanding the different rings could result in new ideas for future oscillator-based photorefractive processors.

We believe that our research group is the first to observe the oscillating patterns occurring within a barium-titanate spherical disk. For now, the oscillating patterns observed inside the spherical disks stand mostly as a scientific curiosity, however, in order to devise possible applications for this phenomenon, it is necessary

to first understand it. In fact, the model presented in section 3.4 suggests that the disk patterns are unidirectional miniature oscillators pumped mostly by a single two-beam coupling interaction, and therefore hold the promise for a future integrated ring processor such as the autotuning filter [Anderson, '92].

As part of a system oriented group, it is natural that within the contributions is that of building and optimizing a photorefractive system for a specific processing task, in this case, that of carrier suppression. The carrier suppressor was developed for a specific application as part of a higher level system, where its function is to remove the correlation between temporal signals which are to be discriminated by the next stage, the autotuning filter. Another application for the carrier suppressor demonstrated in chapter 4 would be in laser tuning. In Ye et al [Ye, '96] a laser beam, whose frequency is to be tuned, is shifted by an acoustooptic modulator and then locked to a reference cavity. In this case, the reference cavity is locked to the ^{87}Rb D_2 line at 780 nm, however the same concept could be used with the I_2 hyperfine resonances at 532 nm [Jungner, '95]. The carrier suppressor used on such a system would allow the use of electrooptic modulators, therefore providing higher modulation bandwidth and, thus, shorter reference cavities which are typically easier to stabilize. The carrier suppressor described in chapter 4 was the evolved result of several other carrier suppressor predecessors. The earlier systems were not covered by this thesis because, first, they didn't accomplish the specifications of our application, and second, all the important aspects of the carrier suppression physics is covered in the description of the last, optimized system. This system rendered a suppression of more than 70 dB, the highest value reported in the literature [Loayssa,

'00; Tonda-Goldstein, '00]. Despite this successful result, there is room for improvement in other aspects of the system. Section 4.4 provides preliminary results in the implementation of a future carrier suppressor which utilizes a channelized electrooptic modulator design. Although the integration provided by this design is attractive for two or three channels, as the number of channels increases, we believe this design will prove impractical due to crystal size and cross-talk issues. In that case, a possible alternative design is that of using a grating to direct the several beams to independent modulators which are, however, placed next to each other on the same mount.

Finally, in chapter 5, we present a closed-form solution for two-beam coupling for the case of only one spatial component on each port. This solution is cast in terms of an operator algebra which was originally developed by Anderson [Anderson, '00; Anderson, '99] and is summarized in section 5.2. We believe that the solution presented is the most general available in the literature as it includes both complex coupling and any number of temporal components. The work of Saxena et al [Saxena, '90] solves the problem of N mutually incoherent pairs of temporal beams. In this case, they show that complete energy transfer between the two write beams may occur in a finite interaction length, as opposed to the infinite thickness required in the standard case of only one temporal component on each beam. This is the same conclusion we arrived at, for example, in the case of the carrier suppression, where the presence of a sideband allowed for perfect suppression at a finite interaction length. However, their solution includes only the case for real coupling, this is, pure energy transfer. The work from Ringhofer et al [Ringhofer, '00] provides the general

complex coupling solution for a phase modulated beam in the gain port, with arbitrary phase modulation strength. However, their solution describes only two temporal components, the carrier and the modulation sidebands, as opposed to an unlimited number of temporal modes as in the solution provided in chapter 5.3. In their paper they also provide an application example, where phase modulation is used in the active stabilization of two-beam coupling by means of an electronically introduced phase feedback. For the case of a single spatial mode on each beam, besides the signal processing applications in our laboratories, such as the demultiplexer [Saffman, '91] or the autotuning filter (previously known as the feature extractor) [Anderson, '92], other applications include RF filtering [Hong, '93], laser bandwidth narrowing [Chomsky, '91] or signal amplification [Hamel de Monchenault, '87]. An application that has currently been brought into the spotlight is that of using heterodyne detection with two-beam coupling for laser ultrasonics [Ing, '91; Puoet, '96; Scruby, '90]. Heterodyne detection with two-beam coupling offers several advantages over the conventional optical heterodyne scheme [Hamel de Monchenault, '88]: First, the signal beam may consist of an arbitrary wave-front, such as speckles for instance, while, in the conventional heterodyne, the signal is limited to a spatial mode which is collinear with the local oscillator plane wave. Second, since the grating adapts to time varying phase distortions which are slow compared to the photorefractive response time. The photorefractive heterodyne detection is well suited for laser ultrasonics, a noncontact method for performing ultrasonic nondestructive evaluation measurements. Unlike the conventional piezoelectric

contact transducers, the noncontact nature of laser ultrasonics allows inspection in

high-temperature or otherwise hostile environments. The method typically consists of

generation of ultrasound with a high-power pulsed laser followed by detection of the

ultrasonic motion of the scattering surface by a photorefractive heterodyne detection

setup (see, for example, [Murray, '00]).

In conclusion, this thesis covers important aspects in the development of

photorefractive system that utilizes two-beam coupling. It covers, first, the

development of component technologies which allow for easy testing of concepts,

second, the development of a specific processing system, the carrier suppressor, and

finally, it provides a solution for the general case of a single spatial mode two-beam

coupling cast in terms of an operator formalism.

BIBLIOGRAPHY

[Anderson, '87].   Anderson, D.Z. and Lininger, D.M., *Dynamic optical interconnects: volume holograms as optical two-port operators*. Appl. Opt., 1987. **26**: p. 5031-8.

[Anderson, '89].   Anderson, D.Z. and Feinberg, J., *Optical novelty filters*. IEEE J. Quantum Electron., 1989. **25**: p. 635-47.

[Anderson, '91].   Anderson, D.Z., et al., *Photorefractive flip-flop*. Opt. Lett., 1991. **16**: p. 250-2.

[Anderson, '92a].   Anderson, D.Z., Benkert, C., and Crouch, D.D., *Competitive and cooperative multimode dynamics in photorefractive ring circuits*, in *Neural Networks for Perception, vol. 2, Computation, Learning, and Architectures*, H. Wechsler, Editor. 1992a, Academic: Boston.

[Anderson, '92b].   Anderson, D.Z., et al., *Optical implementation of a self-organizing feature extractor*, in *Advances in neural-information processing systems IV*, J.E. Moody, S.J. Hanson, and R.P. Lippmann, Editors. 1992b, Morgan Kaufmann: San Mateo, Calif. p. 821-8.

[Anderson, '95].   Anderson, D.Z., Saffman, M., and Hermanns, A., *Manipulating the information carried by an optical beam with reflexive photorefractive beam coupling*. J. Opt. Soc. Am. B, 1995. **12**: p. 117-23.

[Anderson, '99].   Anderson, D.Z., Brockett, R.W., and Nuttall, N., *Information dynamics of photorefractive two-beam coupling*. Phys. Rev. Lett, 1999. **82**(7): p. 1418-21.

[Anderson, '00].   Anderson, D.Z., *A matrix formulation of photorefractive two-beam coupling*. Phys. Rev. A (submitted), 2000.

[Baer, '87].   Baer, T., *Continous-wave laser oscillation in a Nd:Yag sphere*. Opt. Lett., 1987. **12**(6): p. 392-4.

[Baxter, '50]. Baxter, W.T., *Jewelry Gemcutting and Metalcraft*. 3rd ed. 1950, New York: McGraw Hill.

[Breugnot, '95].   Breugnot, S., et al., *Low-noise photorefractive amplification and detection of very weak signal beams*. opt. Lett, 1995. **20**(5): p. 447-9.

[Buse, '97].   Buse, K., *Light-induced charge transport processes in photorefractive crystals. I. Models and experimental methods*. Appl. Phys. B (Lasers and Optics), 1997. **B64**(3): p. 273-91.

[Chang, '90]. Chang, T.Y., Hong, J., and Yeh, P., *Spatial amplification*. Opt. Lett., 1990. **15**(743-745).

[Chomsky, '91].   Chomsky, D., et al., *Laser frequency bandwidth narrowing by photorefractive two-beam coupling*. Opt. Lett, 1991. **17**(7): p. 481-3.

[Cohen-Tannoudji, '77a]. Cohen-Tannoudji, C., Diu, B., and Laloe, F., *Complements AIV and EIV*, in *Quantum Mechanics*. 1977a, Hermann and John Wiley & Sons: Paris, France.

[Cohen-Tannoudji, '77b]. Cohen-Tannoudji, C., Diu, B., and Laloe, F., *Complement EIII*, in *Quantum Mechanics*. 1977b, Hermann and John Wiley & Sons: Paris, France.

[Courtney, '98]. Courtney, D., et al., *Room temperature CW operation of InGaAsP/InP microdisk lasers*. Proc. SPIE – Int. Soc. Opt. Eng., 1998. **3286**: p. 138-51.

[Cronin-Golomb, '84]. Cronin-Golomb, M., et al., *Theory and applications of four-wave mixing in photorefractive media*. IEEE J. Quantum Electron., 1984. **QE-20**(1): p. 12-30.

[Denz, '99]. Denz, C., *Optical Neural Networks*. 1999, FIZ Karlsruhe, Germany: Vieweg; Braunschweig.

[Fainman, '86]. Fainman, Y., Klancnik, E., and Lee, S.H., *Optimal coherent image amplification by two-wave coupling in photorefractive BaTiO₃*. Opt. Eng., 1986. **25**: p. 228.

[Feinberg, '80]. Feinberg, J., *Real-time edge enhancement using the photorefractive effect*. Opt. Lett., 1980. **5**: p. 330-2.

[Feinberg, '82]. Feinberg, J., *Self-pumped, continous-wave phase conjugator using internal reflection*. Opt. Lett., 1982. **7**(10): p. 486-8.

[Feinberg, '83]. Feinberg, J., *Optical image processing using four-wave mixing in photorefractive materials*. Proc. SPIE – Int. Soc. Opt. Eng., 1983. **388**: p. 106-11.

[Feinberg, '90]. Feinberg, J., *Phase conjugation with photorefractive materials*. Opt. & Photon. News, 1990. **1**(12): p. 30-3.

[Gabor, '69]. Gabor, D., *Associative holographic memories*. IBM J. Res. Dev., 1969. **13**(156-159).

[Gu, '93]. Gu, C. and Yeh, P., *Photorefractive devices for optical neural networks*. Optical Memory & Neural Networks, 1993. **2**(3): p. 185-98.

[Gunter, '88]. Gunter, P. and Huignard, J.-P., *Photorefractive materials and their applications*. Vol. I & II. 1988, New York: Springer Verlag.

[Hamel de Monchenault, '87]. Hamel de Monchenault, G., Loiseaux, B., and Huignard, J.-P., *Amplification of high bandwidth signals through two-wave mixing in photorefractive Bi₁₂SiO₂₀ crystals*. Appl. Phys. Lett., 1987. **50**(25): p. 1794-6.

[Hamel de Monchenault, '88]. Hamel de Monchenault, G. and Huignard, J.-P., *Two-wave mixing with time-modulated signal in Bi₁₂SiO₂₀ theory and application to homodyne wave-front detection*. J. Appl. Phys., 1988. **63**(3): p. 624-7.

[He, '94]. He, Q.B. and Yeh, P., *Fanning noise reduction in photorefractive amplifiers using incoherent erasures*. Appl. Opt., 1994. **33**(2): p. 283-7.

[Hong, '90]. Hong, J., Chiou, A.E., and Yeh, P., *Image amplification by two-wave mixing in photorefractive crystals*. appl. Opt., 1990. **39**: p. 3026-9.

[Hong, '93]. Hong, J.H. and Chang, T.Y., *Frequency-agile rf notch filter that uses photorefractive two-beam coupling*. Opt. Lett., 1993. **18**: p. 164-6.

[Horowitz, '91a]. Horowitz, M. and Fischer, B., *Photorefractive novelty filters and the dynamics of their nonlinear wave mixing*. Proceedings of the 17th Convention of Electrical and Electronics Engineers in Israel, 1991a: p. xiv+429, 10-14.

[Horowitz, '91b]. Horowitz, M., Kligler, D., and Fischer, B., *Time-dependent behavior of photorefractive two- and four-wave mixing*. J. Opt. Soc. Am. B, 1991b. **8**(10): p. 2204-17.

[Ing, '91]. Ing, R.K. and Monchalin, P., *Broadband optical detection of ultrasound by two-wave mixing in a photorefractive crystal*. Appl. Phys. Lett., 1991. **59**: p. 3233-5.

[Jahoda, '84]. Jahoda, F.C., Weber, P.G., and Feinberg, J., *Optical feedback, wavelength response, and interference effects of self-pumped phase conjugation in BaTiO₃*. Opt. Lett., 1984. **9**(8): p. 362-4.

[Joseph, '92]. Joseph, J., et al., *Real-time image processing using selective erasure in photorefractive two-wave mixing*. Appl. Opt., 1992. **31**: p. 4769-72.

[Jungner, '95]. Jungner, P., et al., *Absolute frequency of the molecular Iodine transition R(56) 32-0 near 532nm*. IEEE Trans. Instrum. Meas., 1995. **44**: p. 151-4.

[Khoury, '91]. Khoury, J., Woods, C.L., and Cronin-Golomb, M., *Photorefractive holographic interference novelty filter*. Opt. Commun., 1991. **82**(5-6): p. 533-8.

[Khoury, '93]. Khoury, J., Ryan, V., and Cronin-Golomb, M., *Photorefractive frequency converter and phase-sensitive detector*. J. Opt. Soc. Am. B, 1993. **10**(1): p. 72-82.

[Khoury, '94]. Khoury, J., et al., *All-optical joint Fourier transform correlator*. Appl. Opt., 1994. **33**: p. 8216-25.

[Klein, '86a]. Klein, M.B., et al., *Imaging threshold detector using a phase-conjugate resonator in BaTiO₃*. Opt. Lett., 1986a. **11**(575-577).

[Klein, '86b]. Klein, M.B. and Valley, G.C., *Characteristics of BaTiO₃ for electro-optic devices*. Proc. SPIE - Int. Soc. Opt. Eng., 1986b. **567**: p. 116-20.

[Kogelnik, '69]. Kogelnik, H., *Coupled wave theory for thick hologram gratings*. Bell Syst. Tech. J., 1969. **48**(2909-2947).

111

112

[Kukhtarev, '79]. Kukhtarev, N.V., et al., Holographic storage in elctrooptic crystals. II. Beam coupling - Light amplification. Ferroelectrics, 1979. 22: p. 961-4.

[Kuwata-Gonokami, '95]. Kuwata-Gonokami, M., et al., Polymer microdisk and microring lasers. Opt. Lett., 1995. 20(20): p. 2093-5.

[LaGasse, '94]. LaGasse, M.J., et al., Optical carrier filtering for high dynamic range fibre optic links. Electron. Lett., 1994. 30(25): p. 2157-8.

[Lin, '98]. Lin, Y., et al., Observation of the disk mode pattern in organic microdisk. Solid State Comm, 1998. 105(7): p. 445-8.

[Lindsay, '70]. Lindsay, R.B., Lord Rayleigh- the man and his work. 1970. New York: Pergamon Press.

[Lininger, '90]. Lininger, D.M., et al., Theory of bistability and self pulsing in a ring resonator with saturable photorefractive gain and loss. Opt. Commun., 1990. 76: p. 89-96.

[Liu, '93]. Liu, L. and Liu, X.. Matrixing coupled wave theory of photorefractive hologram recorded by two-beam coupling. J. Mod. Opt., 1993. 40(11): p. 2257-65.

[Loayssa, '00]. Loayssa, A., Benito, D., and Garde, R.T., Optical carrier-suppression technique with a Brillouin-erbium fiber laser. Opt. Lett., 2000. 25(4): p. 197-9.

[MacCormack, '96]. MacCormack, S. and Feinberg, J., Revealing 180° domains in ferroelectric crystals by phorefractive beam coupling. Appl. Opt., 1996. 35(30): p. 5961-3.

[Marrakchi, '90]. Marrakchi, A., et al., Dynamic holographic interconnects with ananlog weights in photorefractive crystals. Opt. Eng., 1990. 29(3): p. 215-24.

[Mazur, '99]. Mazur, A., et al., Light-induced charge transport in photorefractive $BaTiO_3$:Fe and $Ba_{0.77}Ca_{0.23}TiO_3$:Fe. Radiat. Eff. Defects Solids (Switzerland), 1999. 150(1-4): p. 673-8.

[Mills, '85]. Mills, P. and Paige, E.G.S., Holographically formed, highly selective, infra-red filter in iron-doped lithium niobate. Electron. Lett., 1985. 21: p. 885-6.

[Min, '88]. Min, K., et al., Automated two speaker separation system. Intern. Conf. on Acoustics, Speech and Sig. Proc., 1988: p. 5 vol. 2928, 537-40 vol.1.

[Mok, '91]. Mok, F.H. Tackitt, M.C., and Stoll, H.M., Storage of 500 high-resolution holograms in a $LiNbO_3$ crystal. Opt. Lett., 1991. 16(8): p. 605-7.

[Montgomery, '95]. Montgomery, R. and DeSalvo, R., A novel technique for double sideband suppressed carrier modulation of optical fields. IEEE Photon. Tech. Lett., 1995. 7(4): p. 434-6.

113

[Murray, '00]. Murray, T.D., Tuovinen, H., and Krishnaswamy, S., Adaptive optical array receivers for detection of surface acoustic waves. Appl. Opt., 2000. 39(19): p. 3276-84.

[Neifeld, '93]. Neifeld, M.A. and Psaltis, D., Programmable image associative memory using an optical disk and a photorefractive crystal. Appl. Opt., 1993. 32(23): p. 4398-409.

[Psaltis, '88]. Psaltis, D., Brady, D., and Wagner, K., Adaptive optical networks using photorefractive crystals. Appl. Opt., 1988. 27(1752-1759).

[Puoet, '96]. Puoet, B.F., et al., Heterodyne interferometer with two-wave mixing in photorefractive crystals for ultrasound detection on rough surfaces. Appl. Phys. Lett, 1996. 69: p. 3782-4.

[Rakuljic, '93]. Rakuljic, G.A. and Leyva, V., Volume holographic narrow-band optical filter. Opt. Lett., 1993. 18(6): p. 459-61.

[Ringhofer, '00]. Ringhofer, K.H., et al., Shaping of photorefractive two-wave coupling by fast phase modulation. Phys. Rev. E, 2000. 61(2): p. 2029-37.

[Saffman, '91]. Saffman, M., Benkert, C., and Anderson, D.Z., Self-organizing photorefractive frequency demultiplexer. Opt. Lett., 1991. 16: p. 1993-5.

[Saxena, '90]. Saxena, R., et al., Diffraction properties of multiple-beam photorefractive gratings. J. Opt. Soc. Am. B, 1990. 7(7): p. 1210-5.

[Sayano, '88]. Sayano, K., Rakuljic, G.A., and Yariv, A., Thresholding semilinear phase conjugate mirror. Opt. Lett., 1988. 13: p. 143-5.

[Schiller, '91].Schiller, S. and Byer, R.L., High-resolution spectroscopy of whispering gallery modes in large dielectric spheres. Opt. Lett, 1991. 16(15): p. 1138-40.

[Schiller, '92].Schiller, S., et al., Fused-silica monolithic total-internal-reflection resonator. Opt. Lett., 1992. 17(5): p. 378-80.

[Scruby, '90]. Scruby, C.B. and Drain, L.E., Laser Ultrasonics, Techniques and Applications. 1990, Bristol, UK: Adam Hilger.

[Sharp, '94]. Sharp, E.J., et al., Photorefractive image processing using mutually-pumped phase conjugators. Proc. SPIE – Int. Soc. Opt. Eng., 1994. 2237: p. 347-59.

[Solymar, '96]. Solymar, L., Webb, D.J., and Grunnet-Jepsen, A., The physics and applications of photorefractive materials. Oxford series in optical and imaging sciences II. 1996, New York: Oxford University Press Inc.

[Staebler, '75]. Staebler, D.L., et al., Multiple storage and erasure of fixed holograms in Fe-doped $LiNbO_3$. Appl. Phys. Lett., 1975. 26: p. 182-4.

[Stojkov, '92].Stojkov, P., Timotijevic, D., and Belic, M., Symmetries of two-wave mixing in photorefractive crystals. Opt. Lett., 1992. 17(20): p. 1406-8.

114

[Tonda-Goldstein, '00].   Tonda-Goldstein, S., et al., *Stimulated Brillouin scattering for microwave signal modulation depth increase in optical links.* Electron. Lett., 2000. **36**(11): p. 944-6.

[Uesu, '95].   Uesu, Y., et al., *Recent development of optical novelty filter and dynamics of optical novelty filter with use of the CAT-type self-pumped phase conjugate mirror.* Ferroelectrics, 1995. **174**(1-2): p. 133-48.

[Volk, '94].   Volk, T., Rubinina, N., and Wohlecke, M., *Optical-damage-resistant impurities in lithium niobate.* J. Opt. Soc. Am. B, 1994. **11**(9): p. 1681-7.

[Weiss, '89].   Weiss, S., et al., *Photorefractive dynamic optical interconnects.* Proc. SPIE - Int. Soc. Opt. Eng., 1989. **1018**: p. 55-7.

[White, '82].   White, J.O., et al., *Coherent oscillation by self-induced gratings in the photorefractive crystal BaTiO3.* Appl. Phys. Lett., 1982. **40**: p. 450-2.

[Ye, '96].   Ye, I., et al., *Hyperfine structure and absolute frequency of the $^{87}RbSP_{3/2}$ state.* Opt. Lett., 1996. **21**(16): p. 1280-2.

[Yeh, '85].   Yeh, P., *Theory of Unidirectional Photorefractive Resonators.* J. Opt. Soc. Am. B, 1985. **2**: p. 1924-8.

[Yeh, '88].   Yeh, P., Chang, T.Y., and Ewbank, M.D., *Model for mutually pumped phase conjugation.* J. Opt. Soc. Am., 1988. **5**: p. 1743-9.

[Yeh, '92].   Yeh, P., *Photorefractive phase conjugators.* Proc. IEEE, 1992. **80**(3): p. 436-50.

[Yeh, '93].   Yeh, P., *Introduction to Photorefractive Nonlinear Optics.* 1993, New York: John Wiley & Sons Inc.

[Yu, '92].   Yu, F.T.S., et al., *Optical novelty filter with phase carrier.* Opt. Commun., 1992. **92**(4-6): p. 205-8.

[Yu, '94].   Yu, F.T.S., Yin, S., and Wang, C.-M., *A content addressable polychromatic neural net using a (Ce:Fe)-doped LiNbO₃ photorefractive crystal.* Opt. Commun., 1994. **107**: p. 300-8.

[Zel'dovich, '95].   Zel'dovich, B.Y., Mamaev, A.V., and Shkunov, V.V., *Speckle-wave interactions in application to holography and nonlinear optics.* 1995, Boca Raton, FL: CRC Press, Inc.

[Zozulya, '95].   Zozulya, A. and Anderson, D.Z., *Spatial structure of light and nonlinear refractive index generated by fanning in photorefractive media.* Phys. Rev. A, 1995. **52**: p. 878-81.

[Zozulya, '95].   Zozulya, A.A., Saffman, M., and Anderson, D.Z., *Stability analysis of two photorefractive ring resonator circuits: the flip-flop and the feature extractor.* J. Opt. Soc. Am. B, 1995. **12**: p. 1036-47.

This thesis entitled:

An Optically-Switched Transmit/Receive Lens Array for Beam-Space Adaptive

Communication Systems

written by James Edwin Vian

has been approved for the

Department of Electrical and Computer Engineering

_____

Zoya Popović

_____

Louis Scharf

Date _____

The final copy of this thesis has been examined by the signatories;

and we find that both the content and the form meet acceptable presentation

standards of scholarly work in the above mentioned discipline

HRC Protocol _____

AN OPTICALLY-SWITCHED TRANSMIT/RECEIVE

LENS ARRAY FOR BEAM-SPACE ADAPTIVE

COMMUNICATION SYSTEMS

by

JAMES EDWIN VIAN

B.S., University of Colorado at Boulder, 1994

M.S., University of Colorado at Boulder, 1996

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

2000

Vian, James Edwin (Ph.D., Electrical Engineering)

An Optically-Switched Transmit/Receive Lens Array for Beam-Space Adaptive Communication Systems

Thesis directed by Professor Zoya Popović

With the growth in todays modern communication world, multiple user and multi-path environments are becoming an ever increasing problem. These environments reduce the quality of the communication link through multiple users interfering with each other or self interference in multi-path environments. A common technique used to combat these environments, is to use a planar antenna array that steer its receptivity pattern in the direction of the desired user. The receptivity patterns for the arrays are control through adaptive algorithms that adjust the magnitude and phase of each received signal at each element in the planar array before coherently combining the signals. The magnitude and phase adjustments (complex weights) reverse the phase shifts induced in the signals as they propagate across the surface of the planar array. The disadvantage of this technique is that it requires significant amounts of computational processing power for large arrays containing many antenna elements. By using a microwave lens array instead of a planar array in the communication system, the received signals are transformed from a phase-space representation to a beam-space representation, which can reduce the processing load for the algorithm and may increase the overall signal to noise ratio (SNR) through partial beam-forming before the noise is added to the system. For these reasons, an optically controlled transmit/receive lens array is developed. The optical control of the lens array allows the array to switch between transmit and receive modes rapidly with negligible interference to the microwave signals. A low optical power single pole double throw switch is developed for routing the transmit and receive signals in the

array that has insertion loss of 0.3 dB and isolation of 36 dB. To model the performance of lens arrays for wireless communication systems, code is developed that calculates the imaging properties of lens arrays under different design conditions. The lens array modeling code is used in conjunction with a modified Least Mean Square (LMS) algorithm that turns off small valued complex weights, to improve the overall SNR and adaptation rate for the communication system.

ACKNOWLEDGMENTS

I would also like to thank the past and present members of Zoya Popović's research group. To Stein Hollung, whose work on transmit/receive lens arrays is the bases for my work research. To Eric Bryerton, Joe Tustin, Michael Forman, Todd Marshall, Manoja Weiss and Jan PeetersWeem, whose conservations on on electromagnetic theory (among other things) help me to better understand this difficult topic. A special thanks to Shawn Stone, Pete Kirkpatrick and Paul Smith, who help with the buying, building and testing of many of the items in this thesis. I would also like to thank Michael Forman, Todd Marshall and Jan PeetersWeem for their time and patience in teaching me the ways of Linux and LaTex which made the writing of this thesis possible.

In addition to the members of Zoya Popović's group, I would like to thank Prof. Dana Anderson, Prof. Kelvin Wagner and their research groups (Edeline Fotheringham, Leslie Czaia, Greg Kriehn, Ken Anderson, Friso Schlottau, Andrew Kiruluta, Paulo Silveria) for there help and guidance in the world of optics. I am also grateful to Prof. Lloyd Griffiths for his help on adaptive arrays and the LMS algorithm. I want to thank our administrative assistants, Helen Frey and Rachel Tearle, who often went out of their way to make sure that our day-to-day operations

DEDICATION

Dedication ... This thesis is dedicated to my family. To parents, Wayne and Sharon Vian, who are my first teachers that made me excited about the world of science. To my sister, Carol Vian, for the healthy competition while growing up. And finally to my loving wife, Trina Vian, who helped me keep everything in perspective.

ran smoothly.

CONTENTS

TABLES

FIGURES

FIGURE

CHAPTER 1

INTRODUCTION

1.1 Adaptive Communication Systems

In recent years there has been an explosion in growth in civilian wireless communications. With the increased use of wireless communications systems, such as cellular phones, and data communications systems, such as wireless local area networks (WLAN), the wireless environment is getting increasingly more complex. With the exponential increase in users and faster data rates, the once plentiful resources of bandwidth (BW) and transmission power are now scarce. In an effort to increase the capacity of communication systems where multiple users and multiple propagation path interference becomes a problem, advanced modulation, coding and power control techniques have been developed, [3, 4, 5, 6, 7]. These techniques have gone a long way towards increasing the systems capacity without sacrificing system performance.

In wireless communication systems, multiple antennas may be used to improve the system performance through diversity transmission and reception. Diversity is effectively transmission and/or reception of multiple copies of the same information that are slightly different. When the multiple copies are combined coherently or non-coherently, the resulting effective signal to noise ratio (SNR) of the system is much greater than that of a single copy. The more copies of the signal the larger the performance improvement of the system (usually measured in bit error rate, BER, decrease).

The concept of diversity dates back to the 1940's when it was used to improve radar performance. Much of the early published work focuses on diversity reception where multi-path fading is a problem [8, 9, 10, 11, 12]. For example, transmission over long distances (i.e., part-way around the world) results in multiple reflections of the waves off the ionosphere, and the different reflected waves interfere with each other at the antenna. When this interference is destructive, it is referred to as "multi-path fading", since the signal fades in that case. Price and Green [12], showed that if delayed copies of the received signals where combined together in what they called a "Rake" receiver, much of the effects of the multi-path fading can be removed. Multi-path fading also occurs in communication systems where the transmission distances are shorter. However, if multiple antennas are used, the fading at each antenna will be different and depends on the distance between antennas. The farther the antennas can be separated, the more uncorrelated the fading at each antenna is. When two antennas are about ten free-space wavelengths (10λ) apart, the multi-path fading at each antenna can be considered independent and the signals may be combined based on their average signal to noise ratio (SNR) [8, 9, 10]. Turin [11] showed that if the fading at each antenna is correlated, then the optimal combination of the received signals is a coherent weighted sum using the mean of the signal fading for the weights, and a non-coherent weighted sum based on the covariance of the the the signal fading for the weights. A completely coherent summation is preferred for these statistical diversity systems, since in this case the SNR grows as the square of the order of diversity, in contrast to linearly for the non-coherent case.

With the advent of the microprocessor it becomes feasible to build antenna diversity systems that use adaptive algorithms to control the combination of the signals received at the different antenna elements. Since the algorithm is able to

perform real time tracking of the received signals, the antenna signals may be co-herently combined depending on the instantaneous relationship between the signals, and not the average off all possible realizations of the correlation between them. As a result, the SNR for an adaptive antenna diversity system is usually better than the SNR for a statistical antenna diversity system. Since the instantaneous relationship between the signals is important, the antennas may be placed at a smaller spac-ing (on the order of $\lambda/2$), making the system more compact. As mentioned earlier, modern communication systems have to maintain quality of service with multiple users interfering with each other, as well as the same user interfering with itself due to multi-path fading. Since adaptive arrays can learn about their environment, they can adapt to the environment without any prior knowledge of the signal sources, assuming that the environment changes slowly when compared to the convergence rate of the algorithm.

One of the first adaptive antenna diversity systems dates back to the 1950's when Howells and Applebuam used a dual-antenna system to remove the effects of an unwanted user, [13]. The magnitude and the phase of one antenna signal is adjusted so that when it is combined with the second antenna signal, the unwanted user is canceled out. This is an example of an adaptive antenna diversity system that isolates users in a multi-user environment. If the second received signal is another copy of the "desired" user's signal due to multi-path, then the magnitude and phase of the antenna signals would be adjusted such that the received signals combine coherently.

The magnitude and phase changes in the received signals are often referred to as weights, in reference to a complex weighted sum. The weights in the algorithm have an interpretation in microwave antenna array theory. The weights are the relative magnitudes and phases of the generators at each antenna element for a transmit array. Because of reciprocity, this is equivalent to the amplitude and phases

that would need to be imposed on the receiver voltages at each element in reception for equal transmit and receive far-field radiation patterns. In an adaptive system, the complex weights vary based on a feedback signal. This in turn changes the far-field radiation pattern of the array in some pre-determined fashion. For example, it places the main beam in the direction of the "desired" user, and a null in the interference direction. If the user is in a multi-path environment, then the weights are adjusted so that there is a lobe pointed in each direction from which the "desired" signal is received. For a multi-user environment, a null is placed in the direction of the "other" user, and very little of its signal is received.

Military applications are also showing an increased need for adaptive arrays both for multi-user communication systems and radar systems. As with the civilian communication environment, military environments are prone to interference from other users and multi-path fading. The military environments also have an additional interferer, "The Enemy", who is jamming "desired" user signals with a signal that can be many times greater in magnitude than that of the "desired" user. If the complex weights are implemented with relative phases, the system is narrowband and results in changes of radiation pattern as the frequency changes. This is often referred to as "beam squint" and is yet an unsolved problem in practical broadband arrays. For broadband signals, a complex weighted sum of time delayed versions of the signals are needed to prevent beam squint. This processing technique is referred to as true-time delay processing.

One of the disadvantages of adaptive array systems is that they require significant computational power to calculate the adaptive weights. The number of computations increases geometrically with the number of array elements. In this thesis, a new type of front end (antenna array and receiver) is explored in the context of adaptive systems. In a "standard" planar array, each element receives a phase-shifted version of the same input plane wave. In the lens array presented in this thesis,

the antenna array performs a Fourier transform and images the signal sources onto a focal surface, and the signal image is then sampled with receivers. Most applications require both transmit and receive functions, and a part of this thesis is concerned with fast efficient routing of signals through transmit and receive paths. To that end, the transmit and receive array presented in this thesis uses optical control which does not interfere with microwave signals and can be very fast and energy efficient.

1.2 Adaptive Planar Array Communication Systems

The basic architecture of a narrowband adaptive array is shown in Figure 7.3. It consists of an array whose signals are adjusted both in magnitude and phase (i.e. each element has a complex weight) before being combined. The weights are controlled by an algorithm which uses information about the structure of the "desired" user's signal (i.e. modulation, coding, pulse shape, etc.) to estimate the instantaneous relationships between the received signals. The information about the "desired" signal is contained in the training signal. There are many different algorithms used in adaptive antennas depending on the needs of the system [13, 14, 15, 16, 17]. One of the earliest algorithms is the LMS (Least Mean Square) algorithm develop by Widrow [18]. It is a gradient search algorithm that minimizes

Figure 1.1. An example of a narrowband planar array adaptive receiver shown for the single user case. The received signals, $s_i(t)$, are amplified with LNAs and down-converted (mixer, local oscillator and LPF) before being sampled (represented by switches). For this model, the effect of the LNAs and down-converters is the addition of noise. The received signals plus noise are sampled with a sampling period T, giving signals, $x_{i,k}$, used by the algorithm. The algorithm uses the training signal, d_k, to adjust the weights, W, such that the SNR at the output, y_k, is maximized.

the error between the output signal, y_k, and the training signal d_k (Figure 7.3). The result is a set of weights that converges to the optimal set of weights known as the Wiener solution [19]. Once they converge, the weights stay fixed until the communication environment changes and the algorithm proceeds to readjust the weights.

The adaptive system in Figure 7.3 is intended for a narrowband application of bandwidth BW, where the delay of the received signals across the array or the difference in delay between multi-path signals is less than $\frac{1}{BW}$. For very broadband systems, such as some used by the military, or for large delay multi-path environments, true-time delay processing is needed. The length of the delay line is at most
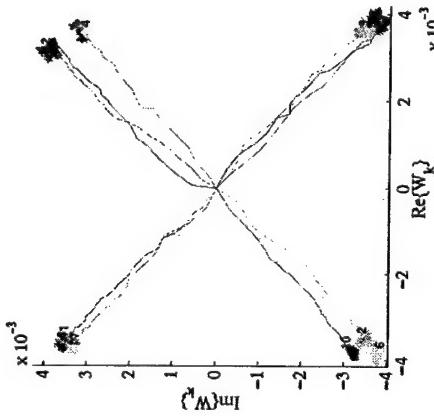
Figure 1.2. The weight trajectories in the complex plane for a planar array adaptive receiver converging to the optimal weights.

twice the maximum delay expected to be seen between signals, and the time interval are spaced at a period of $\frac{1}{BW}$. An entire area of research focuses on this type of adaptive system, referred to as Space-Time Adaptive Processing, STAP [6].

The Wiener solution is the optimal solution for the weights that minimizes the error between the training signal and the output. The Wiener solution

$$W_{opt} = R^{-1}P \qquad (1.1)$$

where $(R)_{ij} = E[x_i x_j^*]$ is the correlation of the received signals, and $(P)_i = E[x_i d]$ is the correlation of the training signal and the received signals. For a standard narrowband planar array, each of the array elements receives approximately the same amplitude (for far-field sources), but with a relative phase shift that depends on the array period and angle of incidence. Using these assumptions in Equation (1.1), the solution results in complex weights that lie on a circle, Figure 7.4. The weight trajectories in Figure 7.4 show the convergence of the LMS algorithm for a plane wave incident on a planar array with ten elements in a row. If an interfering signal is present, the weights rotate so that the power of the interfering signal is

reduced below the noise floor of the system. This weight distribution in the complex plane implies that each of the received signals has equal importance. For adaptive algorithms, the adaptation rate is inversely proportional to the number of weights and the computational load is proportional to the number of weights.

For systems that require small beamwidths (i.e., radar), have large numbers of interfering users and need many nulls, or require multiple beams due to multi-path, a large array with many elements is needed to provide the degrees of freedom for a good solution. In these situations, adaptive planar arrays are slow require large and require significant amounts of computational power [20].

1.3 Adaptive Lens Array Communication Systems

The multi-user, multi-path and jammer environments have one thing in common: they all have signals arriving from different directions. For a planar array, the angle of arrival is encoded in the phase variations across the surface of the array. If a microwave lens is used, the signals are instead imaged onto a focal surface at different locations "behind" the lens. By sampling the focal surface with detectors (antennas and receivers), the algorithm only has to process those detectors illuminated by the "desired" user's signal. The architecture of the lens naturally separates spatially the received signals much like a pre-formed beam space array. If an interfering signal is present, then the weights for the detectors illuminated by the "desired" user's signal adjust to cancel out any of the interfering signal they receive. For multi-path environments, there are several spots illuminated by the "desired" user's signals and the corresponding detectors are used. One obvious advantage of lens arrays is that the number of weights depends on the number of paths the "desired" user's signal takes to the array and not the number of elements in the array. By using fewer weights, the adaptive system should adapt faster and require significantly less computational power.

There are many architectures for microwave lenses. For example, dielectric lenses look and act like optical lenses. Network lenses use beam-forming networks that mimic the beams produced by sampling the image surface of a lens (e.g., Butler matrices). Such networks can be made efficient [21, 22, 23] and are lossless since the beams are orthogonal [24, 25, 26]. However, for large arrays, these networks can have significant loss due to the power combiners and phase shifters contained in them [27]. Finally, there are "Boot lace" lens arrays which are two arrays connected together in element pairs with transmission lines. The transmission lines simulate the delay that a microwave signal undergoes through a dielectric lens. The transmission lines are longer in the center than on the edge, in analogy to an optical lens being thicker in the center. Lens arrays are also prone to loss just as network lenses, but the loss in a array lens grows slower than the loss in a network lens as the number of elements increases, [27].

Lens arrays may have an advantage over planar arrays by performing some beam-forming before noise in added by receivers. A discussion on the effects of lens arrays on communication systems, which points to interesting future work in given in the Section 7.3.2.

1.4 Fast Optical Control of Lens Arrays

Bi-directional lens arrays are are needed for radar and communications systems. However, it is possible to design bi-directional lenses with amplifiers that add gain to the system. The amplifiers are connected between each pair of antenna elements. One of the major advantages of active lens arrays is that they can produce very powerful transmitting signals through spatial power combining, which in turn means that lower-power, less expensive, more efficient power amplifiers can be used in transmission [27].

In a bi-directional array, the transmit signals are routed through power
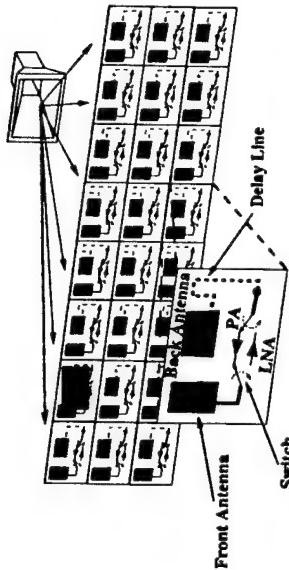
Figure 1.3. A cylindrical bi-directional lens array which uses SPDT switches to route through either the transmit path (PA) and receive path (LNA) respectively.

amplifiers (PAs) and the receive signals are routed through LNAs. This can be accomplished in two ways. The first way is to use circulators that direct signals depending on their direction of propagation. Unfortunately, these devices are too large for use in lens arrays. The second way is to direct the signals using single pole double throw (SPDT) switches, as in Figure 1.3. By using two SPDT switches in each element, the lens array is limited to half- duplex operation, which can be limiting for some communication systems, but not for radar. The rise time and fall time for the switches should be on the order of a few nanoseconds to accommodate wide bandwidth signals. For example, a 2 ns rise time, τ_{rise}, switch can handle a

$$BW = \frac{1.8}{2\pi \tau_{rise}} = 140 \text{ MHz [28]}$$ signal at its fastest switching rate.

In the past, half-duplex bi-directional lens arrays have been electrically controlled [1]. The control lines for the switches weave in and out between the unit cells, as can be seen in the photograph in Figure 1.4. To minimize the number of control lines in the array, the switches for the unit cells are connected in parallel. By connecting the switches in parallel, the RC time constants for the switches add, and this slows down the rise time of the array. An array with hundreds of unit cells can have a switching rise time of a microsecond, even though nanosecond switches are used in each element.

switch. Chapter 3 discusses the development and characterization of an improved optical switch. Chapter 4 discusses the design and implementation of an optically-controlled transmit/receive lens array. Chapter 5 discusses a numerical model for lens arrays designed to explore the tradeoffs in lens array design. Chapter 6 reviews the LMS algorithm and simulations of an adaptive lens array systems with tradeoffs in hardware and required computational processing. Chapter 7 discusses some topics for future directions that are motivated and validated by the work presented in this thesis.

(a)

(b)

Figure 1.4. Example of an electrically controlled bi-direction lens array that uses orthogonally polarized slot antennas (b) for the array elements [1].

Instead of electrically controlling the switches in parallel, it is possible to optically address each individual unit cell. By using optical fibers, the control optical signals are naturally isolated from the microwave signals and the fibers do not interfere with the microwave radiation. Since each switch is individually controlled, the rise time for the array is the same as the rise time for the unit cell, independent of array size.

1.5 Organization

This thesis presents work on applying lens arrays to communication and radar systems. It covers the development of a fast, low-power optically-controlled transmit/receive lens array and simulations of a lens array adaptive communication system. Chapter 2 reviews microwave switch design, the current work done in optical switches and the development of an optically controlled resonant microwave

CHAPTER 2

8 GHZ RESONANT SWITCH AND UNIT CELL DESIGN

2.1 Microwave Switches Background

The principle goal of the research described in this chapter is to design a fast optically controlled, half-duplex transmit/receive active antenna array. One basic component of the array is an optically controlled single pole double throw switch (SPDT) used to route the transmit and receive signals. The design of the SPDT switch consists of microwave circuit, optical circuit and design .

Switches at microwave frequencies in principle work the same way as switches at low frequencies (few hundred MHz and below): the signal chooses the path that has the lower impedance. There are three measures of performance for a switch: insertion loss (IL); isolation (ISO); and return loss (RL). IL is the ratio of power delivered to a load for an ideal switch in its "on" state to the actual switch in it's "on" state. Therefore IL is a measure of the power lost from the load due to the imperfections in the switch. It is usually expressed in positive decibels. Using scattering parameters, IL=1/|s_{21}| for the switch in the "on" state. ISO is the ratio of amount of power delivered to the load for an ideal switch in the "on" state to the amount of power delivered to the load for actual switch in the "off" state. ISO is a measure of how well the switch turns "off" the load and is expressed as 1/|s_{21}| for the switch in the "off" state. Return loss appears only at microwave frequencies and is the amount of power reflected at the switch because the load is no longer ideally matched to the source. It is expressed as 1/|s_{11}| for the switch.
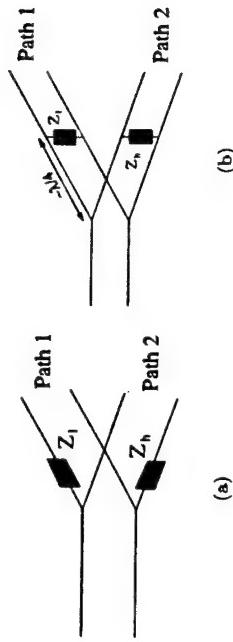
Figure 2.1. Transmission line schematics for a SPDT switch with microwave devices,such as PIN diodes, in series (a) and shunt (b).

The main difference between microwave and low frequency switches is that effective impedance of a device, e.q. a PIN diode, depends on its position in the switch. This leads to two main configurations of microwave switches, the series and shunt configuration, Figure 2.1.

The series configuration operates in the same intuitive way as a low frequency switch with the low impedance device in the "on" path and the high impedance device in the "off" path. The equations for IL, ISO and RL as

$$IL = \left| \frac{(2 + Z_h)(1 + Z_l) + Z_0(1 + Z_h)}{2Z_0(1 + Z_h)} \right|^2 \qquad (2.1)$$

$$ISO = \left| \frac{(2 + Z_l)(1 + Z_h) + Z_0(1 + Z_l)}{2Z_0(1 + Z_h)} \right|^2 \qquad (2.2)$$

$$RL = \left| \frac{Z_{load} - Z_0}{Z_{load} + Z_0} \right| \qquad (2.3)$$

$$Z_{load} = \frac{(Z_l + Z_0)(Z_h + Z_0)}{Z_l + Z_h + 2Z_0} \qquad (2.4)$$

As seen from 2.1-2.3, good switch design requires a device that has a very small impedance in the "on" path and a very large impedance in its "off" path.

As mentioned earlier, in microwave circuits, the effective impedance of a device depends on its position in the circuit. The shunt configuration for SPDT microwave switch exploits this property to improve the IL or ISO of the switch in some cases. For the "on" path, the high impedance state of the device is in

3 MESFETs in parallel

Transmission Lines

Figure 2.2. An example of a MMIC SPDT switch with multiple devices (in this case MESFETs) used to improve IL and ISO.

switch, figure 2.2, uses multiple devices to improve the IL and ISO of the switch over a larger BW. Due to the non-linear nature of the devices, they generally generate large inter-modulation products if two signals are present in the switch at the same time. Table 2.1 shows some typical values of IL, ISO, RL and third-order inter-modulation product (IP3) for commercially available switches.

The recent development of mircoelectromechanical systems (MEMS) has allowed the development of mechanical microwave switches. An excellent tutorial on microwave MEMS switches is given by Elliott Brown, [29], which discusses the advantages, disadvantages and areas of development for MEMS switches. MEMS are mechanical structures built on the micron scale using semiconductor etching techniques developed for integrated circuit. Cantilever and air bridge are the two basic styles for MEMS switch.

The cantilever switch has a conductive arm connected to one side of a gap and suspended over a metal contact on the other side of the gap. To close the switch,

Table 2.1: Examples of commercially available MMIC switches

manufacture	M/A-Com	M/A-Com	M/A-Com	Alpha
part number	2954-2004	MASW20000		AS018R2-00
IL (dB)	1.8	2.1		2.2
ISO (dB)	50	50	42	42
VSWR	2	1.8		1.85
BW (GHz)	2-18	DC-18	DC-18	DC-18
Switching speed	20 ns	2ns	2ns	3ns
Inter-modulation Intercept Point	N/A	43dBm		46dBm

parallel with the intrinsic transmission line impedance (Z_0), resulting in an combined impedance approximately equal to Z_0. Thus, there is a small reflection at the device for sufficiently large impedance devices. For the "off" path, the low impedance of the device shorts the transmission line, causing a large reflection at the device. To prevent the short in the "off" path from shorting out the "on" path, a $\lambda/4$ section of transmission line is used, to transform the short to high impedance (open) at the Y-junction. The shunt configured switch has inherently smaller bandwidth than the series configured switch due to this $\lambda/4$ section of transmission line.

To solve for the IL, ISO and RL for shunt configured switch, $\left[\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right]$ calculated at the Y-junction and used to find IL, ISO, and RL as:

$$IL \text{ or } ISO = \frac{1}{4}\left|A + \frac{B}{Z_0} + CZ_0 + D\right|^2 \qquad (2.5)$$

$$RL = \frac{A + B - C - D}{A + B + C + D}, \qquad (2.6)$$

where

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ Y_1 & 1 \end{bmatrix} \begin{bmatrix} 0 & \frac{j}{Z_0} \\ jZ_0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ Y_2 & 1 \end{bmatrix}, \qquad (2.7)$$

$$(2.8)$$

and Y_1, Y_2 are give for the IL and ISO, respectively, as

$$IL: \qquad Y_1 = \frac{Z_l \| Z_0}{Z_0^2} \qquad Y_2 = \frac{1}{Z_h} \qquad (2.9)$$

$$ISO: \qquad Y_1 = \frac{Z_h \| Z_0}{Z_0^2} \qquad Y_2 = \frac{1}{Z_l}. \qquad (2.10)$$

There are two basic categories of devices use in microwave switches. The first categories is active devices like diodes and transistors. The second categories is mechanical structures that use moving metal contacts.

Active microwave switches have been around for decades and are available as microwave monolithic integrated circuits (MMICs). A typical MMIC SPDT

an electric field is applied between the metal arm and the pull down electrode in the gap. The electrostatic forces generated pulls the arm down making contact with the metal contact on the other side of the switch. The pull-down electrode usually has an insulator deposited upon it to prevent the shorting of the control voltage. By reducing the electric field to a value typically less than the threshold voltage needed to close the switch, the switch will open. This switch is commonly used with microstrip transmission line architectures.

The air bridge switch lends itself to co-planar waveguide (CPW) architectures, where the air bridge is connected between the two ground strip of the CWP air bridge switch and is suspended over the center line. Again, by applying an electric field between the air bridge and an electrode on the center line (not shown), the air bridge bends down to short out the CPW transmission line. In this case the switch path is "off" when the electric field is applied. Then by reducing the electric field, the air bridge opens and turns the switch "on".

The IL, ISO and RL for cantilever and air bridge MEMS switches depends on the difference between the "close" and "open" capacitance between the arm (or bridge) and the transmission line. For good IL in cantilever switch and ISO in the air bridge switch, the capacitance should be as large as possible or preferably an ohmic contact. For a good ohmic contact, strong electrostatic forces are needed to compress the contacts together. For good ISO in a cantilever switch and IL in the air bridge switch, the capacitance should be as small as possible, leading to a large gap between arm (or bridge) and transmission line. MEMS switches can exhibit IL of 0.4 dB below 1 GHz to just below 1 dB up to 40 GHz and ISO of 40 dB below 1 GHz which degrades about 20 dB at 40 GHz.

There is a natural tradeoff between switch rate, performance and control voltage. In order to make a fast switch, the movement of the pieces should be minimized, which reduces the IL or ISO of the switch. By increasing the electric field

strength and the stiffness of the arm (or bridge), the arm is forced to the contact gap. The electrostatic forces generated pulls the arm down making contact with the metal contact on the other side of the switch. The pull-down electrode usually has an insulator deposited upon it to prevent the shorting of the control voltage.

By reducing the electric field to a value typically less than the threshold voltage needed to control the switch, the switch will open. This switch is commonly used with microstrip transmission line architectures.

quicker and relaxes faster. It appears that the MEMS switches are fundamentally limited to a "close" time of about 1 μs and a "open" time of about 10 μs due to material properties.

Since electrostatic forces are used to control the MEMS switch, the required energy needed to control the switch equals the difference in the stored energy in the "open" and "close" states. The total control power is the stored energy times the switch rate, $\frac{1}{2}(C_{close} - C_{open})V^2 f_s$. For example, with "close" capacitance ("open" capacitance is negligible) of 13 pF, a close state voltage of 4 V and a switch rate of 10 kHz, the total power dissipated is approximately 1 μW. Besides low control power, MEMS switches also display superior linearity from the lack of non-linear devices like diodes and MESFETs.

2.2 Photonic Switches Background

The earliest photonic switches were developed at much lower RF frequencies [30, 31]. The switches consist of a gap in a conductive path filled with photo-conductive materials. The switch is "closed" when the photo-conductive material is illuminated with watt-level optical power. Much of the practical work focuses on RF and microwave photonic switches that are fast [32, 33, 34]. These switches are used in high-power inductive energy storage pulse power systems [35]. One reported switch is used to control a 100 kW power source with a PIN diode and 2D laser diode array [36]. Only in the past five to ten years has work been done on microwave photonic switches. Most of the effort has been focused on developing a PIN diode that acts as good switch diode as well as a good photodiode. This is one technique for developing a microwave photonic switch, two more techniques are presented in section 2.3.1.

By using a PIN diode as the microwave and the optical device, the optical

signal provides the power for control; no additional power from supply bias line is needed. The optical signal generates the carriers in the PIN diode, reducing the diode impedance. The more light illuminates the device, the lower the devices impedance.

Therefore, the device is in a high impedance state with no light, and in the low impedance state with high optical power density.

A device that has good optical performance and microwave performance is not easy o obtain. From the microwave point of view, the PIN diode should have a small junction capacitance for a high "off" impedance and low "on" resistance. From the optical point of view, the PIN diode should have a large active area for more light gathering ability and use materials that have good responsivity (generation of carriers from photons). These conditions oppose each other in the optimization of the PIN diode design. For example, a large active area causes a large junction capacitance. Low "on" resistance means more optical power, or large active area, and good responsivity materials. Materials used in PIN photodiodes that have good responsivities typically have larger loss than the materials used for high quality microwave PIN diodes.

In [37], demonstrated a switch is presented with IL=2.2 dB for 1 mW of optical power and with rise and fall time of .01 μs to 10 μs depending on optical power. The IL is limited by the mount of optical power delivered to the PIN diode, while the ISO is limited by the diode capacitance. [38] demonstrated a switch with IL of 1.2 dB for 40 mW of optical power and ISO of 30 dB. Both switches need on the order of milliwatts of optical power to achieve good IL with good ISO.

2.3 Developing Photonic Switch

As presented in the previous section 2.2, most of the work done in photonic switches has been done in order to develop devices that could sever both as a microwave and optical device. In this work, we use off-the-shelf devices for both the

microwave and optical circuits. Most of the fast optical components are develop for the optical communication industry, limiting the wavelengths to 880 nm, 1310 nm and 1500 nm, which are all in the infrared (IR).

Besides being limited to off-the-shelf components, the ability to machine parts is limited to what can be manufactured on a printed circuit board (PCB) prototyping milling machine model 93s made by LPKF. The milling machine can cut, drill and gouge substrate materials, FR4 (PCB material), brass and aluminum of up to 1.6 mm (64 mils). It has an accuracy of 8 μm (0.3 mils) with a minimum drill hole size of 300 μm in diameter. The minimum cutting round is 250 μm (10 mils) for substrates, 1 mm for FR4 and 2 mm for metal.

One of our goals was also to minimize the size ofthe switch and the required optical power. A small switch size is needed it insure proper array element spacing to prevent grating lobes. Low optical power is needed not only for overall efficiency, but also in order to be able to control large arrays with a single standard laser-diode.

2.3.1 Techniques for Optical Control The previous work shown in section 2.2 demonstrated only one means of optical control of a microwave switch. In principle there are three techniques for optical control. The first is to have a optoelectronic device like a photodiode or photo-transistor act as the microwave device that switches between the high and low impedance state, Figure 2.3(a). This is the technique used in section 2.2. The second is to use a photo-voltaic device to provide the bias power for a microwave device, which changes impedance states Figure, 2.3(b). For example, a PIN photodiode which generates carries when illuminated and therefore biases a microwave PIN diode. The third technique uses a optoelectronic device to control the bias current for a microwave device, Figure 2.3(c). Each technique has its own advantages and disadvantages, and has been investigate to some degree in this project.

The first technique uses a optoelectronic device in a microwave switch. This

(a)

(b)

(c)
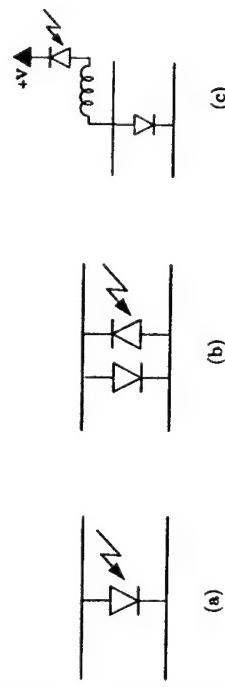
Figure 2.3. Three principle ways to optically control a microwave switch: use a photo-device as a microwave device in the switch (a); use the photo-device to power the microwave device in a switch (b); and use the photo-device to control the bias power to a microwave device (c).

technique requires the fewest number of devices and supply bias lines, however it is difficult to produce an optoelectronic device that functions well as both a microwave and optical device(see section 2.2). Another disadvantage of the this technique is the IL and ISO are a function of optical power.

The second technique uses the photo-voltaic device to bias a microwave device, which in turns provides the high and low impedance for the microwave switch. Again, this type of control does not require external bias for the microwave switch, but still requires substantial optical power (10's of mW). By using a microwave PIN diode in the switch, the IL and ISO equal that of a traditional microwave switch. Since the microwave PIN diode "on" impedance depends on bias current, the IL or ISO for this technique depend on the optical control power.

The third technique uses an optoelectronic device to control the bias voltage for the microwave device. This technique also has the advantage of good IL and ISO, as well as the potential of requiring much less optical power, however it requires an external supply bias line for the switch. If the switch is integrated with microwave amplifiers or other active components that require bias power, then the photonic switch does not necessarily add extra bias lines.

2.3.2 Investigating Optical Devices Used as Microwave Switch

Components The first technique for optical control uses an optoelectronic device as the microwave device in a microwave switch, which is ideal due to the small numbers of optoelectronic and space required. Several PIN photodiodes with active areas ranging from 300 μm to 80 μm are tested in series and shunt configurations in hopes of being able to use the first type of optical control. Unfortunately, there is little difference (few dB) between the IL and ISO for the "on" and "off" states of the PIN photodiodes. In section 3.1.1 there is evidence presented that points to the bond wires used in the packaging of the photodiode leading to the poor IL and ISO performance. There is also significant loss (between 3 dB and 6 dB) due to the resistive nature of the photodiodes.

2.3.3 Investigating Optical Devices That Bias Microwave Switch

Component The second technique of optical control using a photo-voltaic device to power a microwave device has the advantage in not needing an external power source [39, 40]. This technique is very difficult to test in the laboratory because of the difficulty in coupling large amounts of optical power onto the active area of the device. Most microwave PIN diodes require about 1 mA to 3 mA of current at 1 V to 1.5 V and it is not easy to generate enough current at the required voltage from a photodiode. Several photodiodes can be connected in series to produces the required voltage. A typical responsivity for infrared InGaAs photodiodes is 0.95 A/W, implying that 3 mW of optical power is needed at each photodiode to generate enough current to bias the microwave PIN diode "on".

Eight PIN photodiodes connected in series to control a microwave varactor diode with 450 μW of optical power has been demonstrated [41, 42]. The success of this design relies heavily on the fact that reverse bias current for a varactor diode is only a few μA. The complexity of delivering large amounts of optical power to multiple photodiodes per switch makes this technique impractical for array applications.

Figure 2.4. Three optoelectronic device used for control of microwave PIN diode: SFH300 photo-transistor (a), LI4F1 photo-Darlington (b) and TSL251 photo-amp (c). Note that the total circuit dimension is roughly one free-space wavelength at 10 GHz.

PIN diode with very little optical power.

A true photo-MESFET would use the optical signal to generate carriers in its gate region. Without the ability to produce specialized devices such as photo-MESFETs, a circuit version is developed in this work, Figure 2.5. The photodiode controls the gate bias voltage on the MESFET, thus turning it "on" or pinching it "off". The circuit form of a photo-MESFET does require an additional negative voltage bias line to pinch off the MESFET.

Two photo-MESFETs are developed using the components given in Table 2.3. The first design uses the EG&G photodiode, but we later switched to the Fermionics photodiode for its smaller package with negligible change in performance. As mentioned in section 2.1, MESFETs can be used as the "high" and "low"

Figure 2.5: Schematic for the circuit implementation of a photo-MESFET

Table 2.2: Performance data for devices in Figure 2.4

manufacturer	Siemens	Texas Instruments	QT Optoelectronics
type	Photo-transistor	Photo-amplifier	Photo-Darlington
part number	SHF 300	TSL250	LI4F1
wavelength	880 nm	840 nm	850 nm
Irradiance Responsivity	2-4 (mA/mW/cm²)	80 (mV/μW/cm²)	3 (mA/mW/cm²)
Rise/Fall Time	10 μs	90 μs	250 − 300 μs
Collector Current	50 mA	1.6 mA	

2.3.4   Investigating Optical Devices That Control the Bias of Microwave Switch Component   The third technique uses an optoelectronic device to control the bias current to microwave device. If a photodiode is used for the optoelectronic device, then 3 mW of optical power is still required to generate enough carriers for the 3 mA current to flow through the photodiode. This is a lot of optical power for control, especially when bias power is available. Photo-transistors and photo-amplifiers have much larger effective responsivity through the use of devices with gain, such as transistors or operational amplifiers. Two photo-transistors and one photo-amp, Table 2.2, are tested with a microwave PIN diode. All three of the devices are able to control the PIN diode with very little optical power (μW), however the response time for the devices ranges from 10 μs to 250 μs. This is on the order of what is possible for electrical control for large arrays, which does not warrant the extra cost and complexity of optical control. The physical size of the devices also prohibits their use in microwave arrays, Figure 2.4.

2.3.5   Photo-MESFET   The photo-amps in sections 2.3.4 use MOS-FETs devices for gain which are inherently slow compared to microwave standards. Using the photo-amp concept, but a MESFET instead as the gain device with a faster response time, the resulting photo-MESFET should be able to control the

(a)

(b)

Figure 2.6. The measured RL (a), IL (blue –) and ISO (red --) (b) for a photo-MESFET in a shunt configuration

Table 2.3. The performance data of the MESFET and two photodiodes used to implement the photo-MESFET

	HP AFT-26836	EG&G C30637ECER	Fermionics FD80S3
Type	General Purpose	Active Area (dia)	80 μm
		75 μm	
Gain	9 dB	Responsivity	0.85 A/W
		0.86 A/W	
Saturated Drain Current	30 mA	Capacitance	0.4 pF
		0.4 pF	
Bandwidth (GHz)	16	Bandwwidth (GHz)	2
	16	3.5	2
		3.5	

impedance devices in microwave switches resulting in a simpler switch design than having the photo-MESFET control another microwave device. The shunt configured photo-MESFET switch performs the best at 9.1 GHz with an IL of -2.3 dB, ISO of -19.2 dB and bias current of 30 mA, Figure 2.6. A RL of -1.37 dB when the channel is "off" and -11.7 dB when the channel is "on" shows that there is 1 dB to 2 dB of power loss in the MESFET itself.

The 30 mA of current required by the photo-MESFET is undesirable for use in the switch, so the photo-MESFET is tested with microwave diodes. Two of the diodes used with the photo-MESFET are given in Table 2.4 and have surprisingly similar results compared to the photo-MESFET. Like the MESFET, the diodes give the best IL and ISO in a shunt configuration. The HSMP3892, Figure 2.7, has IL of -4.4 dB and ISO of -17.4 dB at 9.08 GHz for a bias current of 30 mA. The MA4E2054, Figure 2.8, has an insertion loss of 4.2 dB and ISO of -20.2 dB at 9.12 GHz for a bias current of 3.3 mA. The MA4E2054 diode has the same performance as the other photo-MESFET.

Table 2.4. Performance characteristics for two microwave diodes tested with the photo-MESFET.

Manufacture	Hewlett Packard	M/A-Com
Type	PIN Diode	Schottky Diode
Part number	HSMP-3892	MA4E2054
Capacitance	0.30 pF	0.35 pF
"on" Resistance	2.5 Ω	10 Ω

Figure 2.8. The measured RL (a), IL (blue −) and ISO (red - -) (b) for a MA4E2054 in a shunt configuration

Figure 2.7. The measured RL (a), IL (blue −) and ISO (red - -) (b) for a HSMP-3892 in a shunt configuration. All of the s-parameter measurements were maded using HP 8510C Network Analyzer.

(a)

(b)
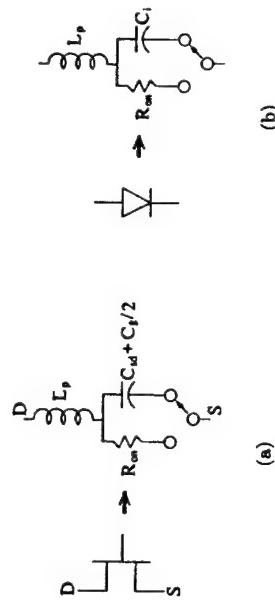
Figure 2.9. Lumped element models for the "on" on "off" characteristics for a MESFET (a) and a diode (b).

devices with only 3.3 mA of current, and is the device chosen for the SPDT switch to be used in the array.

2.3.6 Resonant Switch Operation The similar performance between three very different devices (MESFET, PIN diode and Schottky diode) hints towards an unusual mode of operation for the devices in the switch. All three devices have similar IL and ISO of about -20 dB in a very narrow bandwidth around 9.1 GHz. Another interesting anomaly is the fact that the switch is "off" when the device is "off" and vise versa for a shunt configuration, which is opposite to the theory presented in section 2.1.

The narrow bandwidth of the ISO suggests a resonant operation of the device. The small circuit models for the devices and their packages, Figure 2.9, show that a resonance can occur between the junction capacitance of the device and the bond wire inductance in the package when the device is "off". The series resonance of the inductor and capacitor forms a very low impedance that shorts out the transmission line turning the switch "off".

When the device is "on", the bond wire inductance is in series with the "on" resistance of the device. Estimating that the bond wire inductance of about 0.5 nH for a 0.5 pF junction capacitance resonates at 9 GHz, the resulting IL is about
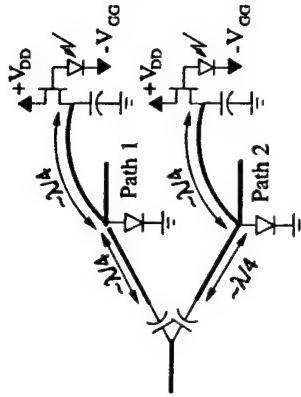
Figure 2.10. Schematic for an optically controlled SPDT switch with PIN diodes in a shunt configuration. Thick lines represent transmission lines.

1.2 dB. This is about 1 to 2 dB better than measured, suggesting some additional "ohmic" loss. The RL for this model is about 3.5 dB which is consistent with measurements for the three devices. Obviously, the package bond wire inductance limits the performance of the switch and a chip PIN diode would perform better. M/A-Com MA4GP032 chip PIN diodes were purchased, but would take 10 weeks to deliver, so work is done on a resonant switch design in the interim as a learning and backup process.

2.4 SPDT Switch Design

The design of a SPDT switch is more elaborate than presented in section 2.1. Care must be taken to isolate the switching bias currents for the two diodes and to suppress loading of the microwave signal by the diode bias circuitry (i.e. photo-MESFET). Series capacitors are used to separate the switching bias currents for the two diodes Figure 2.10. A 70 Ω microstrip line (the highest impedance line that we can mill) connects the photo-MESFET to the diode. A 100 pF capacitor is in shunt with the microstrip line $\lambda/4$ away from the diode, which presents a high impedance to the diode for its microwave short. The initial development of the SPDT switch is done with the HSMP3892 diode, but latter was changed to the MA4E2054 diode
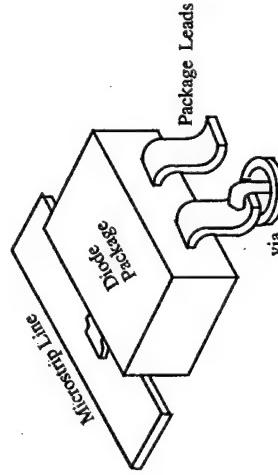
Figure 2.11. Illustration of diode connected to microstrip line with annular ring around the ground via

for its lower current level performance.

2.4.1 Manufacturing Tollerances

A robust repeatable switch design is necessary for success in arrays that contain hundreds of these switches. The first SPDT switch constructed with the HSMP3892 PIN diode demonstrated the sensitivity of the switch to manufacturing variations. The resonant frequency of the diode is very sensitive to how they are connected to the microstrip line. For example, two diodes from the same lot are used in a switch. Both are shorted to ground using a wire, but one has an annular ring, Figure 2.12. The resonant frequency of the switch with the annular ring moved from 9.08 GHz to 8.56 GHz, Figure 2.12(b).

Another experiment was performed to investigate diode placement, Figure 2.13. Since the diode impedance is reflected to the Y-junction, its placement helps determine the RL for the switch as well as the operation frequency. By moving the diode 0.3 mm away from the Y-junction the resonant frequency moves from 8.9 GHz to 8.7 GHz, Figure 2.14(b). The ISO and IL stays relatively unchanged at approximately 23 dB and 5.4 dB respectively.

In order to insure good placement of the diodes, notches are cut into the microstrip lines for the diodes Figure 2.15. To test the effectiveness of the notches a set of diodes are measured in the circuit, remove and then replace back into the
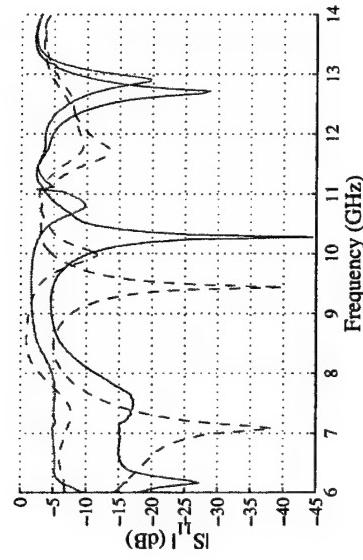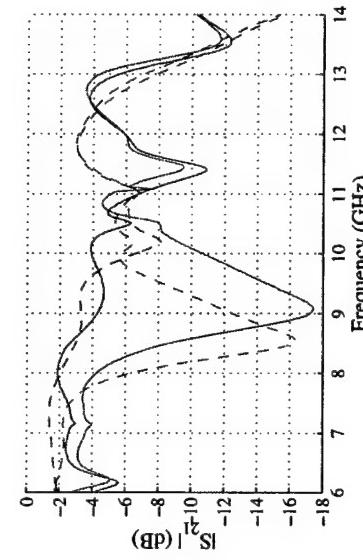
Figure 2.12. Measured RL (a), IL (blue) and ISO (red) (b) for a HSMP-3892 PIN diode in a switch with shunt configuration without (–) and with an annular ring around the ground via (--).

Figure 2.14. The measured RL (a), IL (blue) and ISO (red) (b) for the SPDT in Figure 2.13 before (–) and after (--) the new diode placement.

Figure 2.13. Layout of the SPDT switch illustrating the change in placement for the diode along the microstrip transmission line.

circuit. The overall resonant frequency of the switch did not change, however the ISO was reduced by 2 dB from 19.3 dB to 17.4 dB and the IL changed from 7.4 dB to 3.2 dB, Figure 2.16(b). There are two possible sources for the change. First, the soldering, de-soldering, and re-soldering may have done some thermal damage to the devices, effecting their performance. Second, the solder bubble connecting the diode to the microstrip line has a different shape each time and probably filled the notch gap better in the second set of measurements.

Another experiment examines the switch sensitivity to different sets of diodes. The measurements resulting fro placing a new set of diodes shows a 0.16 GHz change in resonant frequency between the old and new sets of diodes from 8.72 GHz to 8.56 GHz, Figure 2.17(b). The ISO improved from -19.3 dB to -21.0 dB, and the IL improved from -7.4 dB to -4.8 dB. These simple experiments show that even with meticulous care in fabricating the SPDT switch, it will be difficult to construct several switches with similar enough performance. This becomes a problem when the resonant frequency of the switch needs to be the same as the resonant frequency of the patch antenna in each element of the array.

(a)

(b)

Figure 2.16. Measured RL (a), IL (blue) and ISO (red) (b) for diodes placed using notches as shown in Figure 2.15. The same set of diodes are measured in the switch (–), then removed and replaced (--)

Figure 2.15. An illustration of a diode connected to a microstrip line using a notch for percise placement.

2.4.2 Four Different SPDT Switch Designs Using a resonant diode in a SPDT switch proves to be a difficult to simulate and predict with any accuracy, therefore a shot gun approach to the problem is adopted. Four different SPDT switches are made with different lengths of transmission line between the MA4E2054 Schottky diodes and the Y-junction Figure 2.18.

From the experiments in section 2.4.1, variations in the diodes, placement and soldering resulted in switches that worked at frequencies lower than the 9.1 GHz response predicted in section 2.3.6. The four SPDT switch designs have incremental changes in the layout based on a switch design centered at 8 GHz. Theoretically, the distance between the diode and the Y-junction should be $\lambda/4$. The four switches designated "a", "b", "c" and "d" use transmission line lengths of 0.8 $\lambda/4$, 0.89 $\lambda/4$, $\lambda/4$ and 1.07 $\lambda/4$. From the IL and ISO measurements, circuit "a" has the best performance with a resonant frequency at 8.15 GHz, IL=3.6 dB, ISO=24.6 dB and RL=6.3 dB, as shown in Figure 2.19.

2.4.3 Effects of Optical Control Until now, most of the effort focuses on the RF performance of the switch and how it relates to the diodes, setting the optical control aside. The optical control part of the SPDT switch requires a photo-MESFET to control the diode current and optical mounts to align fibers to the photodiode, as shown in photograph Figure 2.20. To understand the effects of the

Figure 2.18. The general layout of optically controlled SPDT switch for the circuit in Figure 2.10. The placement of the diode (distance d) is adjusted to optimize the SPDT switch performance.

optical control on the SPDT switch, the "a" SPDT switch is tested under different stages of construction. The first stage has just the Schottky diodes and capacitors, with the bias control provided through a bias tee. The second stage has MESFETs added with their gate bias being controlled. The third stage adds the optical mount. The optical mount is made out of FR4 material with a 3 mm minimum spacing between the FR4 and the microstrip lines. Finally, the photodiode is added and the switch is controlled using a laser-didoe. At each stage of the experiment, IL and ISO are affected with a net result of 2-3 dB increase in IL and ISO, as shown in Figure 2.21.

2.5 Unit Cell Array Element Design

The unit cell design, Figure 3.1, is essentially two SPDT switches connected

---
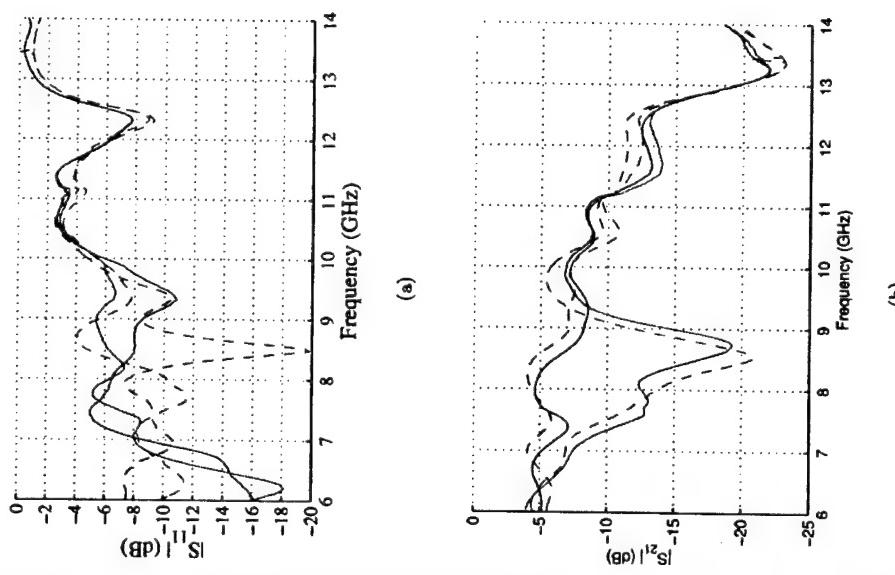
Figure 2.17. Measured RL (a), IL (blue) and ISO (red) (b) for two sets of diodes (– and - -) in a SPDT switch that uses notches for placement.
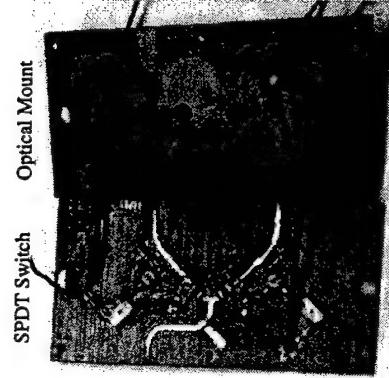
Figure 2.20. Photograph of Optically-controlled 8 GHz SPDT switch with optical mount

back-to-back with the power amplifier (PA) and low noise amplifier (LNA) connecting them. The photo-MESFET circuit is modified to contain a push-pull photodiode pair for rapid turning "on" an "off" of the MESFET [43]. Each MESFET in turn provides the bias control for the appropriate pair of PIN diodes in the two SPDT switches. The switches and MMICs are powered through a 5 V supply line with a 1 kΩ current-limiting resistor for the switches. The two antennas are connected to the Y-junction ports of the SPDT. Due to the amount of circuitry in the unit cell, a substrate with a high dielectric constant is needed to shrink the overall circuit layout. Rogers' Duriod 6010.5 with a thickness of 0.508 mm (25 mils) is used.

2.5.1 Low Noise Amplifier and Power Amplifier

The MMIC amplifiers, Table 2.5, are chosen for three practical reasons. They are low cost, available with short delivery times, and require only a single supply voltage. Both MMICs have input and output matching filters that reject the lower frequency switching current, Figure 2.23. The matching filters are designed assuming gold bond wires that are 0.254 mm (10 mil) long and have a diameter of (0.7 mils). The MMICs are mounted on a brass pedestal to minimize total bond wire lenght, Figure 2.24.

(a)

(b)

(c)
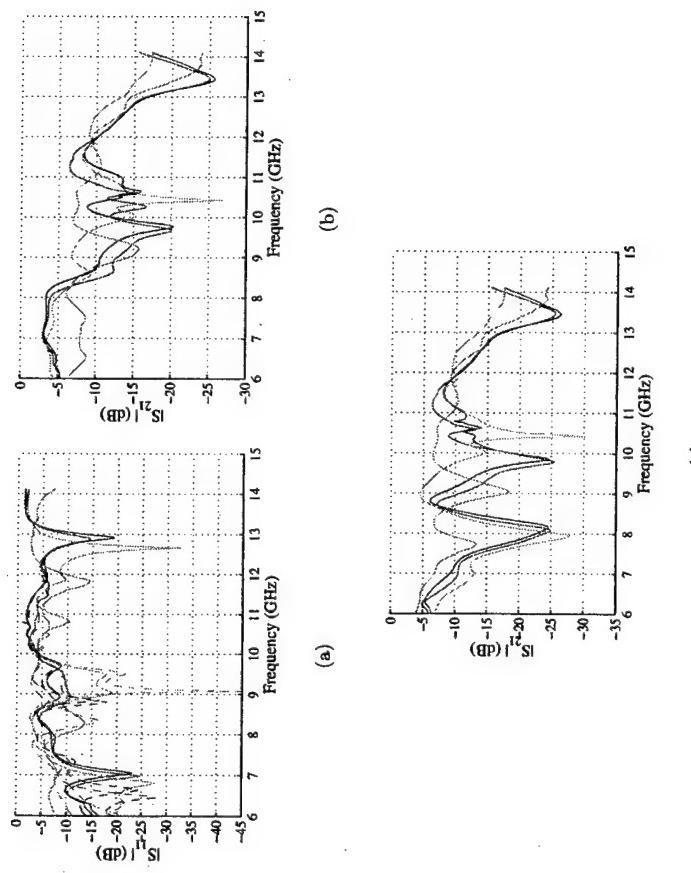
Figure 2.19. The measured RL (a), IL (b) and ISO (c) for four different SPDT switch designs illustrated in Figure 2.18: d=0.8 λ/4 (blue), d=0.89 λ/4 (red), d=λ/4 (green) and d=1.07 λ/4 (yellow). All four circuits are measured with the same calibration.

Figure 2.22. Unit cell schematic with push-pull photodiodes for faster control. Thick lines represent transmission lines.

Figure 2.23. Measurement of RL for LNA at lower frequencies (< 1 Ghz) demonstrating the isolation of the LNA from the diode circuitry.
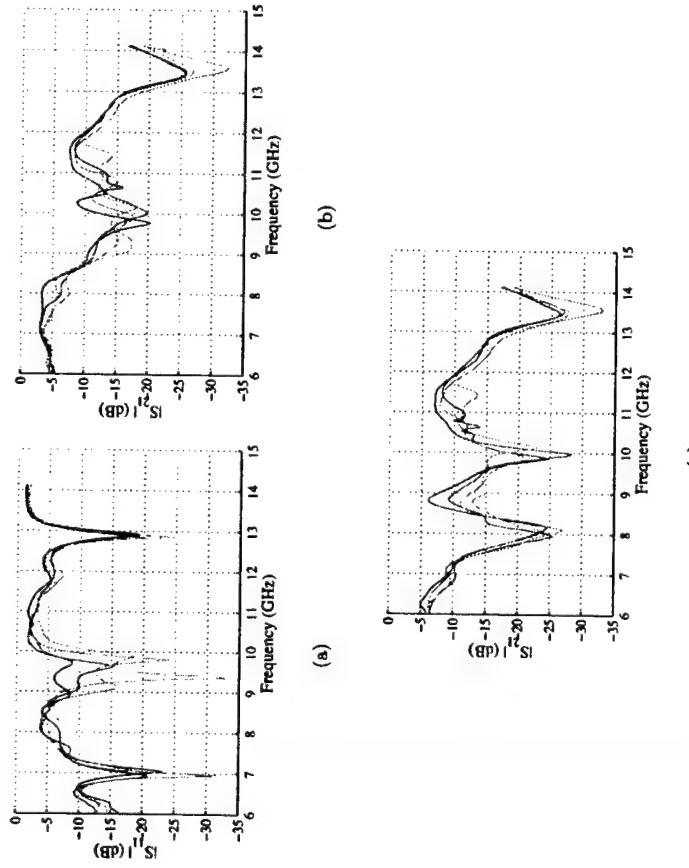
Figure 2.21. The measured RL (a), IL (b) and ISO (c) for d=0.8 λ/4 SPDT switch design (blue), with MESFET (red), with MESFET and optical mount (green), and with optical mount and photodiode (yellow).

Table 2.5: Performance data for the LNA and PA

Manufacture	Hewlett Packard	United Monolithic Semiconductor
Part number	HMMC-5618	CHA2036
Type	Power Amplifier	Low Noise Amplifier
Gain (dB)	14	16
BW (GHz)	6-20	7-13
1 dB Compression (dBm)	18	10
Supply Voltage (V)	5	5
Noise Figure (dB)	5.5	2

Figure 2.26. Measured IL for a 25 mm long filter bias line without 30 gauge wire (blue −) and with the wire (red - -).

bias line. To increase the cross sectional area of the bias line, a 30-gauge wire is soldered down the center of the line. Since the high frequency current density is largest at the edges of the line, the wire should have minimal effect on RF filter performance. The measurements of a 25 mm long filter bias line designed at 8 GHz shows about 10 dB suppression of 8.5 GHz signals without the wire, and about 8 dB with the wire, Figure 2.26.

2.5.3 Antenna Design

Two types of printed antennas are used most often in planar lens array designs: slot antennas and microstrip patch antennas. Slot antennas consist of a slot cut into a ground plane and fed with a microstrip line(Figure 2.27) or a CPW. Slot antennas radiate on both sides of the ground plane in a hemispherical pattern with nulls in the direction of the ground plane [45]. For this reason, when slot antennas are used in planar lens arrays, polarizer are needed to isolate the two sides of the array [44, 1]. For lens arrays used in an angle diversity communication system, the addition of the polarizers would add unnecessary aberrations to the lens, reducing the communication systems performance.

Microstrip patch antennas consist usually of a rectangular piece of metal above a ground plane. They are fed using microstrip lines (Figure 2.28), coaxial

1.2 mm   0.8 mm

(a)   (b)

Figure 2.24. Photographs of LNA (a) and PA (b) resting on brass platforms and biased through 100pF chip capacitors.

2.5.2 Bias Line Design

Supplying power to all the MMICs and switches in an array proves to be a difficult problem and has led to array instabilities [1, 44]. The instabilities arise from free-space coupling of the microwave signal onto the power bias lines. To minimize the coupling to the bias lines, a filter bias line is used. A filter bias line has a cascade of high and low impedance transmission lines that are $\lambda/4$ long at the design frequency, resulting in a notch filter, Figure 2.25. A problem with filter bias lines is their small cross-sectional area for the high impedance sections (assuming microstrip lines), which increases the resistance of the

Figure 2.25. Photograph of filter bias line with a 30-gauge wire for reduced resistance.

lines or aperture feeds. The microstrip patch antenna radiates due to fringing electric fields at the edges of the patch, in a hemispherical pattern on one side of the ground plane with nulls in the ground plane direction [45]. Using microstrip patch antennas in a planar lens array eliminates the need for polarizers. Unfortunately, microstrip patch antennas are resonant and have bandwidths on the order of 2–4%. The bandwidth can be increased using electrically thick substrates at the cost of radiation efficiency due to substrate modes [21].

A 8 GHz non-radiating edge fed patch antenna is designed using a method of moments simulator *Ensemble* by Ansoft. The patch is 5.47 mm by 7.82 mm with

Figure 2.27: Illustration of a microstrip fed slot antenna.

Figure 2.28. Photograph of 8 GHz patch antenna on 10.5 dielectric substrate. Accompanying the patch antenna are five FR4 apertures used to measure the effects of FR4 mount on the patch radiation

Figure 2.29. The measured RL for the patch antenna as it radiates through different size mounts: no mount (blue), 20 mm by 20 mm (red), 17.5 mm by 17.5 mm (green) and 15 mm by 15 mm (yellow). The FR4 mounts with apertures 20 mm by 20 mm or larger have negligible effect on the antenna radiation.

the microstrip feed 0.76 mm from the center of the non-radiating edge. *Ensemble* is used to estimate the coupling between the patch antenna and a microstrip line in its vicinity. In order to maintain 30 dB isolation between the patch antenna and nearby microstrip transmission lines to prevent oscillations, it was found that all lines should be 3 mm from the radiating edge and 2 mm from the non-radiating edge of the patch antenna.

Besides the patch antenna coupling to other parts in the circuit, the patch antenna needs to radiate through a aperture in the optical mount. Since the FR4 material used in the optical mount is lossy at 8 GHz, by measuring the return loss of the patch antenna as the antenna radiates through different apertures, we can determine the effect of the mount. As the FR4 material begins to load the near field of the patch antenna, the return loss should decrease. Five different apertures were built: 15 mm by 15 mm, 17.5 mm by 17.5 mm, 20 mm by 20 mm, 22.5 mm by 22.5 mm and 25 mm by 25 mm, (Figure 2.28). The two apertures smaller that 20 mm by 20 mm reduce the reflected power, as seen in Figure 2.29. A 20 mm by 20 mm aperture is the smallest aperture that the patch antenna can radiate through unchanged. Radiation

Figure 2.30. Measured radiation patterns for the 8 GHz patch antenna alone (blue) and radiating through 20 mm by 20 mm aperture

patterns were taken with and without the FR4 aperture and are shown in Figure 2.30. The aperture attenuates the patch radiation pattern on the sides and the ripples are due to multi-path effects in the test setup (desicribed later in section 4.3.2).

2.5.4 Slot Coupler vs. Vias Feed   With a patch antenna on each side of the unit cell and a ground plane separating them, it is necessary to devise a means of coupling signals from one side of the unit cell to the other. One technique uses a slot coupler, Figure 2.31, which has been shown to have coupling losses of 2 dB and operating up to 25 GHz [46]. Early design work in slot couplers at

Figure 2.31: Illustration of microstrip coupler made from two pieces of substrate.

Figure 2.32: Measured IL for 10 GHz slot coupler on 10.5 dielectric substrate.

10 GHz produced simulated results of coupling loss of 2.76 dB with a RL of 34 dB, showing that slot couplers are prone to substrate modes, Figure 2.32. The measured results for the slot coupler has 0.5 dB of coupling loss and 22 dB of RL at 10.5 GHz. Unfortunately, in order to produce this good performance, meticulous care is taken to align the slots and to ensure good electric contact between the two substrate ground planes, using silver epoxy, without any air gaps. By applying pressure to the slot coupler to squeeze the air gaps the coupling loss changes from about -3 dB to the 0.5 dB. In addition, to the fabrication problems, the currents in slot couplers generally extend several substrate thicknesses beyond the slot, reducing available real-estate. For these reasons, the slot coupler is impractical for the design of the unit cell. Instead, a 0.5 mm diameter brass rod is used as a via to connect the two sides of the array with about 1 dB coupling loss, Figure 2.33.

2.5.5 Unit Cell Layout   The layout for the unit cell follows from the layout of the SPDT switch. The MMICs are placed between the two back-to-back SPDT switches with 0.38 mm (15 mil) brass platforms to raise them up from the center ground plane, Figure 2.34(a). The push-pull photodiodes are connected to the MESFET and the -2.3 V bias line running horizontally across the top and bottom

Figure 2.35. Photograph showing the optical mount placed on the unit cell with the optical fibers.

of the unit cell. The 5 Volt bias line runs vertically on the passive side of the array which also contains one of the antennas, Figure 2.34(b). The optical mount contains the optical fibers and the 20 mm by 20 mm aperture through which the other patch antennas radiates through, Figure 2.35.

2.5.6 Testing Unit Cell The gain of the unit cell is tested in the setup shown in Figure 2.36 for both transmit and receive modes. In both cases, no measurable power was transmitted through the unit cell. As mentioned before, there

Figure 2.36. Illustration of the test setup used to measure the unit cell. The test setup is calibrated to an aperture the size of the unit cell.

Figure 2.33. The measured loss for a 0.5 mm diameter via connecting two microstrip lines on 0.64 mm, $\epsilon_r = 10.5$ substrate material.
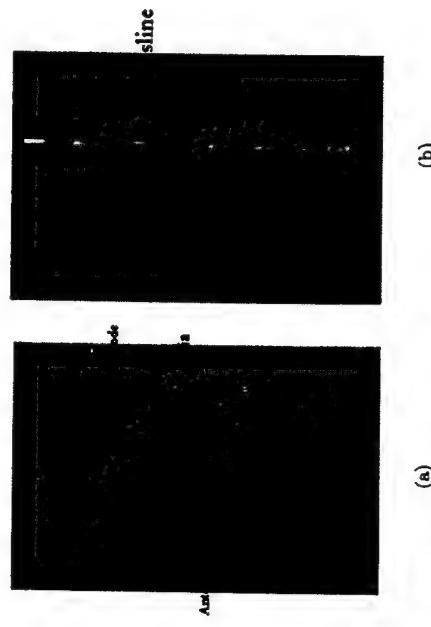
Figure 2.34. Photograph showing the active (a) and passive (b) sides of the unit cell

CHAPTER 3

10 GHZ CHIP-PIN DIODE SPDT SWITCH

3.1 Improved Switch Design

In the previous chapter, a resonant switch was described. This architecture is narrow bandwidth and sensitive to parameter variations. Since the ultimate goal is to integrate this circuit into an array, a design is required which can be manufactured to behave the same many times. Figure 3.1 shows an improved unit cell design. The series and shunt capacitors used to isolate the diodes and provide a microwave short for the bias line have been replaced with high pass filters (HPF) and low pass filters (LPF), respectively. In addition to the push-pull photodiodes, there are push-pull MESFETs for rapid turning on/off of the PIN diodes. Complementary optical pulses control each set of photodiodes. For example, when the transmit photodiodes are illuminated (and the receive photodiodes are not), MESFETs labeled 2 and
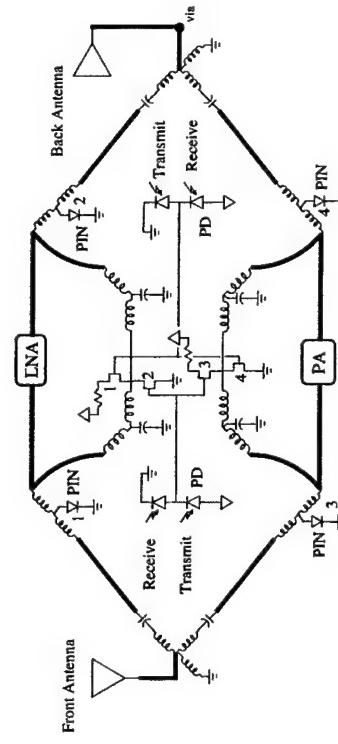
Figure 3.1. Improved unit cell design with push-pull photodiodes and MESFETs, chip PIN diodes, HPFs and LPFs.

Figure 2.37. Measured differential on/off gain for the two unit cells containing only LNA (blue) and PA (red). The noise in the measurement is due the dynamic range of the HP 70820A.

are many difficulties in developing a unit cell with a resonant switch and microstrip patch antenna such that their resonant frequencies overlap. Instead, two unit cells are made, one for transmit mode and one for receive mode, with the Schottky diodes removed. The differential on/off gain of each unit cell agrees with the predicted gains of the MMIC amplifiers at 16.2 dB for the LNA and 14.9 dB for the PA, Figure 2.37. It is also interesting to point out that the patch antennas radiated at different frequencies even though they were made identically and at the same time. This demonstrates the sensitivity of the patch design to manufacturing deviations. A new wide bandwidth switch design is to needed overcome the problem of patch antenna variations.

(a) (b) (c)

Figure 3.2. Three different configurations for connecting a chip diode into a switch: series (a), shunt (b) and T-network (c).

3 in Figure 3.1 are quickly pinched "off" while MESFETs 1 and 4 are turn "on". MESFET one sources current for PIN diodes 1 and 2, while MESFET four discharges the junction capacitance of PIN diodes three and four. With PIN diodes 1 and 2 biased "on", the LNA channel is switched "off", and when the PIN diodes, 3 and 4 biased "off", the PA channel is switched "on".

3.1.1 MA4GP032 Chip PIN diode

As demonstrated in Chapter 2 the package inductance of the diodes and MESFETs limits their switching performance at X-band. The M/A-Com MA4GP032 chip PIN diodes which have an "on" resistance of $1.5\,\Omega$ and junction capacitance of $0.12\,\text{pF}$, are used in the new switch design. There are three configurations for the PIN diode in a switch, (series, shunt and T-network), Figure 3.2. Using lumped element models for the bond wires (inductors) and PIN diode (resistor for "on" and capacitor for "off"), MatLab calculations predict the IL and ISO for the series configuration to be 1.8 dB and 1.7 dB, and for the shunt configuration to be 1.2 dB and 0.95 dB at 10 GHz. The measured IL and ISO are consistent with the lumped element model, showing 0.59 dB IL and 2.1 dB ISO for the series configuration and 0.3 dB IL and 2.3 dB for the shunt configuration, Figure 3.3. However, the T-network configuration performance is greatly improved through reactance cancellation with the bond wires inductance.

There are three design parameters that affect the performance of a PIN
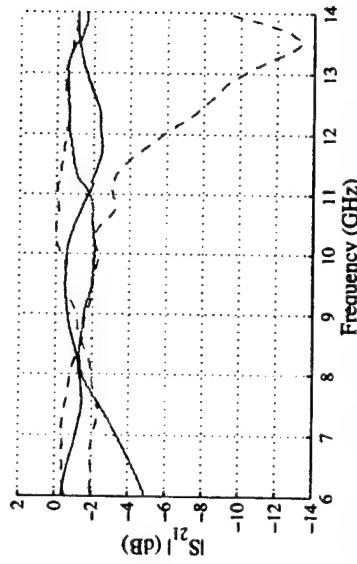
Figure 3.3. Measured IL (blue) and ISO (red) for the series (–) and shunt (--) configurations.

diode SPDT switch in: bond wire inductance (L), distance between diode and Y-junction (d) and transmission line impedance (Z_0). MatLab code based on transmission line equations and lumped element models examines the design tradeoffs for these parameters. Previous measurements of the series and shunt configured PIN diode switches support the use of a lumped element model for predicting IL, ISO and RL. The bond wire inductance is probably the most fabrication sensitive parameter in the switch circuit at 10 GHz. Using a thin microstrip line on a dielectric substrate with $\epsilon_r = 10.5$ as the model for the bond wire above a ground plane, the inductance is estimated to be $2\,\text{nH/mm}$. Depending on the arc of the bond wire and where it connects to the microstrip line, this number can vary by as much as 50%.

Two MatLab simulations of SPDT switch are performed varying the transmission line impedance and length with the bond wire inductance held at 1 nH and 1.5 nH. The IL, ISO and RL is calculated for the switches paying close attention to the sensitivity of switch performance as the parameters change. The lower limit for the inductance at 1 nH is set by the size of the PIN diode. The PIN diode sits on a 0.5 mm diameter brass rod via, limiting the gap in the microstrip line to 1 mm.

From the simulations, the IL and ISO are relatively insensitive to changes transmission line impedance and length, as for the 1 nH case in Figures 3.4. In general, the IL and ISO improves with the increase of transmission line impedance. The RL is the only parameter that significantly sensitive to design parameters, Figure 3.5. Overall, 1 nH has better performance than the 1.5 nH, however caution is warranted since the transmission line equations only give general trends of performance and neglect many high order effects.

To verify the tradeoff analysis, five different circuits are built to test the effects of bond wire inductance and transmission line impedance, Figure 3.6. The first two circuits ("a" and "b") have the PIN diode in a T-network with 1.5 nH and 1 nH bond wire respectively. The second set of circuits ("c" and "d") were built to examine the loss of exponential of $\lambda/4$ and $\lambda/2$ tapers from $50\,\Omega$ microstrip lines to $70\,\Omega$ microstrip line. The final circuit ("e") combines the $\lambda/4$ taper with 1 nH bond wire to measure the total effect of a high transmission line impedance with PIN diode.

The ISO of the switch needs to be sufficient to prevent oscillations in the ring formed by two SPDT switches in the final array element design. In other words the sum of the sum of the loop gain and losses around the loop must be less than the twice the ISO:

$$G_{LNA,dB} + G_{PA,dB} - 2IL_{dB} - 2ISO_{dB} = 0, \tag{3.1}$$

where $G_{LNA,dB}$ and $G_{PA,dB}$ are gains of the LNA and PA respectively. Equation 3.1 that implies a ISO of 15 dB is needed for a switch with 30 dB loop gain (i.e. LNA = 16 dB and PA = 14 dB) to prevent oscillations assuming IL=0. Circuit "b" with 1 nH bond wire inductors has the best performance with IL=0.75 dB and ISO=20.3 dB, which is close to or better than the simulated values of 2 dB IL and 20.2 dB ISO 3.7. Circuit "b" also exhibits low loss as seen in Figure 3.8. The loss from the exponential impedance tapers does dominates over any benefit the higher impedance transmission
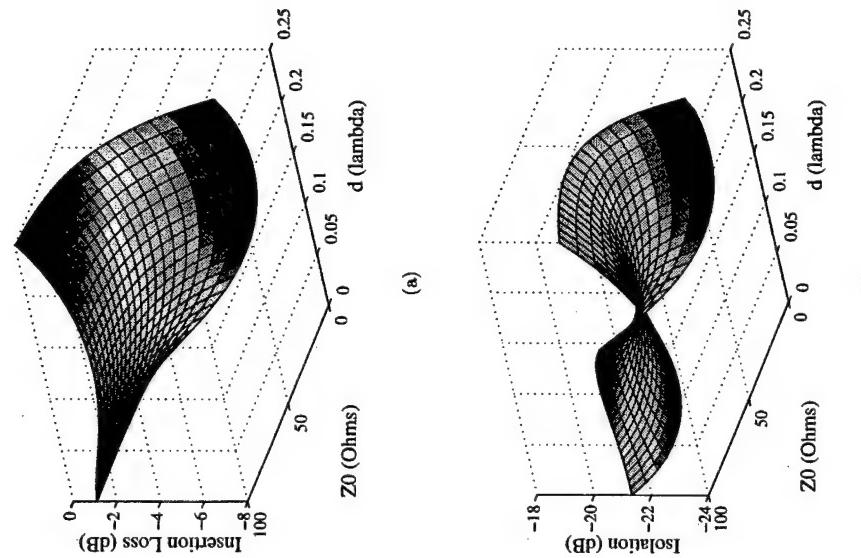
Figure 3.4. Simulated IL (a) and ISO (b) for T-network SPDT switch using MA4GP032 PIN diode with 1 nH bond wires

Figure 3.7. The measured IL (–) and ISO (- -) for circuit "a" (blue), circuit "b" (red) and circuit "e" (green).

Figure 3.8. Measured loss in circuit "b" for the switch "on" (red) and the switch "off" (blue).

Figure 3.5. RL for T-network SPDT switch using MA4GP032 PIN diode with 1 nH bond wires

Figure 3.6. Five test circuits use to examine the performance of PIN diode in different configurations. Circuit "a" is T-network PIN diode with 1.5 nH bond wires and 50 Ω transmission line. Circuit "b" is T-network PIN diode with 1 nH bond wires and 50 Ω transmission line. Circuit "c" is λ/4 impedance taper from 50 Ω to 70 Ω transmission line. Circuit "d" is λ/2 impedance taper from 50 Ω to 70 Ω transmission line. Circuit "e" is T-network PIN diode with 1 nH bond wires and 70 Ω transmission line.

Figure 3.11. Picture of HPF with 2 nH shunt bond wire, 0.25 nH series bond wire and 1 pF chip capacitor.

is adjusted such that the inductor is resonant with the capacitor at 10 GHz. The smallest gap that can be milled on the available PC–board prototyping machine of 5 mils (.0127 mm) creates a 0.25 nH inductor which is resonant with a 1 pF chip capacitor at 10 GHz, Figure 3.11. An addition of a 2 nH shunt inductor increase the attenuation of lower frequency components (below 1 GHz) that are generated by a nanosecond switching currents. The HPF has good performance of 0.1 dB loss at 10 GHz and 20 dB rejection for frequencies below 1 GHz, Figure 3.12.
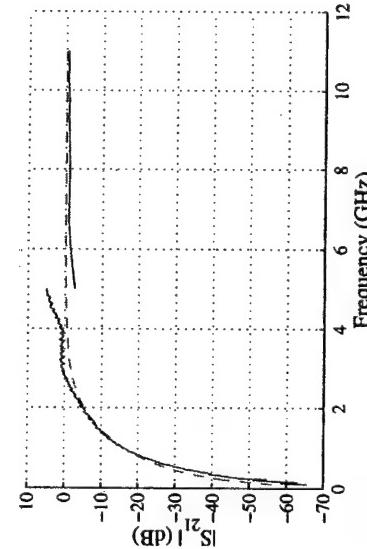
Figure 3.12. Measured (blue –) and simulated lumped element model (red --) performance of HPF. Measurements above 5 GHz are calibrated and uncalibrated below 5 GHz.

Figure 3.9. Measured IL for circuit "c" (blue), circuit "d" (red) and circuit "e" (green).

lines gives in IL and ISO, Figure 3.9.

3.1.2 High Pass Filter

In designing the switch circuit, it is important to make all components very compact in order to have a unit element of the array smaller than a free-space squared. We therefore use lumped elements to design the components. A HPF, Figure 3.10, made from a chip capacitor and bond wire inductors replaces the series capacitor used in the previous SPDT switch design, Figure 2.10. The addition of the series inductor is necessary to bridge the gap in the microstrip line. To minimize the effects of this inductor, the gap distance

Figure 3.10. Circuit schematic (a) for HPF constructed from two bond wires and chip capacitor (b).

(a)

(b)
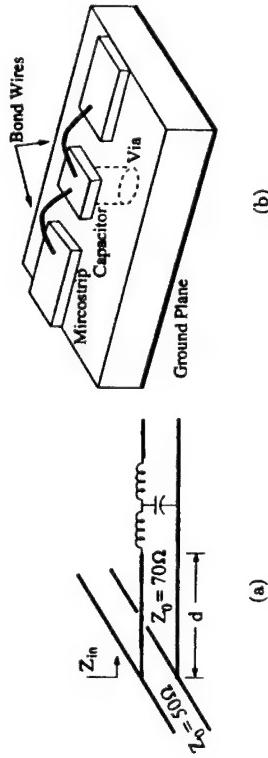
Figure 3.13. Lumped element schematic for LPF constructed from chip capacitor and two bond wires used in biasing the chip PIN diodes (a) and implementation of circuit (b). The distance 'd' that LPF is placed from the 50Ω is adjusted it minimize reflections at the T-junction.

3.1.3 Low Pass Filter

An inductor capacitor LPF, Figure 3.13, replaces the shunt capacitor that suppresses the microwave signal from the diode bias circuits. The 3rd order LPF provides suppression of the 10 GHz carrier while allowing the sub-gigahertz switching currents to pass.

Lumped element models are used to explore effectiveness of three different LPF designs based on three commercially available capacitor values of 1 pF, 3 pF and 5 pF. The LPF with inductance of 0.85 nH and capacitance of 3 pF provides the best compromise between rejection of RF carrier and filter bandwidth. The 3 pF capacitor is also insensitivity to reflection distance as shown in Figure 3.14 with and optimal distance of 58°. A mistake in recording the optimal values for simulations done with *Puff* and *MatLab* led to designs with reflection line distances of 90° and 58°. This mistake turns out to be advantages with the SPDT switch is latter optimized. Testing both circuits, Figure 3.15, shows that the 90° reflection line has the best performance, Figure 3.16. The 90° bias line has a loss of 0.107 dB on the 50Ω microstrip line at the T-junction and 34 dB suppression of a 10 GHz signal through the LPF.
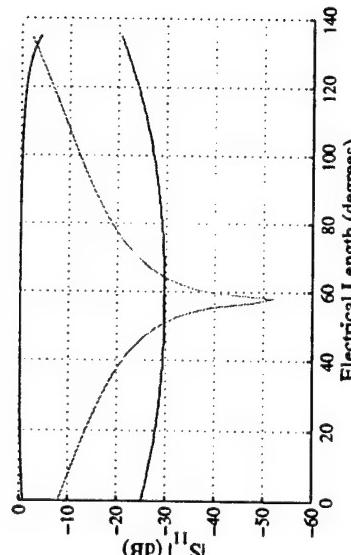
Figure 3.14. Simulated IL (blue), RL (green) for 50Ω microstrip line and LPF isolation (green) as a function of the distance between the LPF and microstrip.

Figure 3.15. Picture showing the two different LPF designs, one with a reflection distance of 90° and the other with a reflection distance of 58.8°.
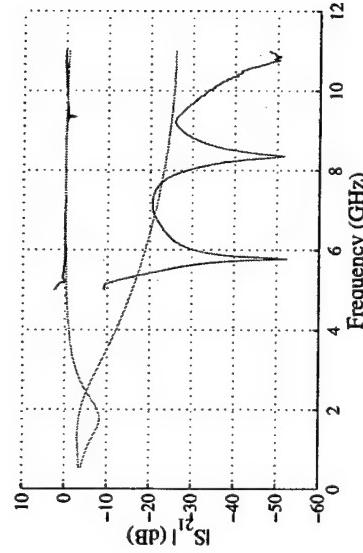
Figure 3.16. Measured IL (blue) and LPF isolation (red) for the LPF design with reflection distances of 90°. The loss in the 50 Ω microstrip line is in (blue) and the LPF rejection is in (red). The simulated performance is in (green).

3.1.4 Simulation of Transient Response of SPDT Switch

A circuit model developed in *SPICE* by Orcad, based on physical measurements of the various components, and combinations of the components is used to model the transient response of the switch. Measured DC voltages and currents for various interconnections of the PIN diodes and MESFETs, Figure 3.17, are used to develop the *SPICE* component models. The photodiode model is based on manufacture specifications. The photodiode model itself uses a voltage-controlled current source to simulate the generation of carriers by optical signals, Figure 3.18.

Figure 3.17. Three different test configurations used to measured DC voltages and currents for developing a *SPICE* model.
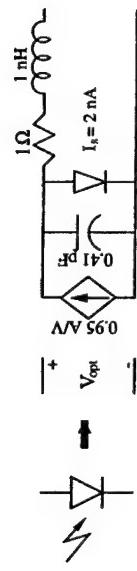
Figure 3.18. A lumped element model for photodiode using a voltage controlled currents source to simulate the generation of carriers as a result of an optical signal.

The rise and fall times for the back-to-back SPDT switch are dependent on the optical power as shown in Figure 3.19. The fastest rise time is 2.4 ns with 9 mW of optical power, limited by the RC time constant for the current limiting resistor and the sum of the PIN diode's capacitances. For optical power smaller than 9 mW, the RC time constant of the MESFET gate capacitance and the "on" resistance of the photodiode dominates. At 1 μW of optical power, the rise time for the switch is 1800 ns. The current limiting resistor increases the rise time as compared to the fall time. It turns out that the photodiode's capacitance effect on the rise and fall times is minimal, and a photodiode with four times the active area and twice the capacitance for a would be acceptable and would on the other hand greatly improve coupling of optical power.
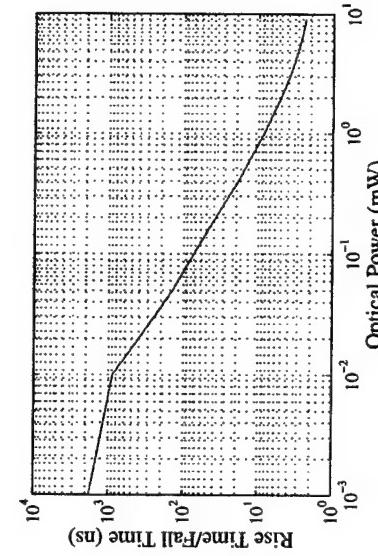
Figure 3.19. *SPICE* simulation of Rise time and fall time for two back-to-back SPDT switches used in the unit cell.

(a)

(b)

Figure 3.21. The measure RL (a), IL (blue b) and ISO (red b) for the second optimized SPDT switch

Figure 3.20. Picture of SPDT switch using MA4GP032 chip PIN diodes with HPFs and LPFs.

3.2 Optimized SPDT Switch Design

Two different approaches are taken in optimizing the design of the SPDT switch from its optimized components. The first design uses *PUFF* to optimize the PIN diode, LPF and HPF designs together, while the second design fixes the filter designs and optimizes only the PIN distance from the Y-junctions, Figure 3.20. The first design optimizes the PIN diode distance (d1) 54° and LPF distance (d2) to 67°. The second design optimizes the PIN diode distance (d1) to 52°. The first circuit has a IL of 1.05 dB (1.55 dB simulated), ISO of 28 dB (-21.8 dB simulated) and 2:1 VSWR BW of 4.64 GHz from 6.48 GHz to 11.12 GHz. The second circuit out performed the previous with IL of 0.315 (1.81 simulated), ISO of 36.33 dB (20.6 dB) and 2:1 VSWR BW of 2.49 GHz from 8.27 dB to 10.76 dB, Figure 3.21.

When measuring the SPDT switches, the HPFs were placed in backwards and shorted the PIN diodes with their shunt inductors. These inductors were re-moved for the measurements shown in Figure 3.21. It ensure that HPF orientation does not affect the SPDT switch performance two Y-junction circuit are built with the HPFs place forwards and backwards as shown in Figure 3.22. The IL and ISO

Figure 3.23. Measured calibrated (–) and uncalibrated (- -) IL for the Y-junction with the HPFs placed correctly and the shunt inductors remove one by one. The measurements with two inductors are in (blue), one inductor are in (red) and no inductors are in (green).

measurements for the two circuits show that both circuits perform the same within a few tenths of dB in IL and few dB in ISO.

3.2.1 Estimation of Switching Current Isolation The SPDT switch has its best performance when the shunt inductors in the HPFs are removed. However, by removing the shunt inductors, will the HPFs affectivily isolate the to sides of the SPDT switch at fast switch rates? From the *SPICE* model, the fastest rise time for the switch was found to be 2.4 ns. A voltage waveform for the PIN diodes with 2 ns rise and fall times, magnitude of 1.2 V and pulse period of 5 ns is constructed in *MatLab* as shown in Figure 3.24(a). This waveform represents the fastest reasonable switch rate for the SPDT switch. The voltage waveform is filtered through the measured HPF's frequency response shown in Figure 3.23. The Fourier transform of the voltage waveform shows that most of its spectral energy is below 500 MHz (Figure 3.24(b)), where the HPF (without shunt inductors) attenuates about 10 dB. The voltage waveform seen by the other PIN diode after passing through the two HPFs is about a tenth of its orginal magnitude. Since the PIN diodes turn "on" at 1 V, the HPFs with shunt inductors do isolate the two sides of

Figure 3.22. The Y-junction circuits used to measure the effect of the HPFs placed forwards (a) and backwards (b).

Table 3.1. Measured input and output power for back-to-back SPDT switch showing maximum input power for the SPDT switch is at least 17 dBm.

Input Power (dBm)	"on" Path Output Power (dBm)	"off" Path Output Power (dBm)	Difference
-5	-30.3	-65.3	-34.9
0	-25.6	-61.6	36.0
5	-20.7	-57.1	36.4
10	-15.6	-51.8	36.2
16.7	-9.1	-44.4	35.31

the SPDT switch at its fastest switch rate.

3.2.2 Maximum Input Power for SPDT Switch

The maximum microwave input power for the SPDT switch is when the PIN diodes rectify enough of the microwave signal and self-bias. When this occurs, the IL and RL will increases and the difference between the IL and ISO will decrease. The SPDT switch is tested with RF input power ranging from -5 dBm to 17 dBm while measuring the outputwave forms for the "on" and "off" paths. The output power for the two paths increases linearly with input power for the full range, while maintaining a 36 dB difference between the IL and ISO, Table 3.1.

3.3 Switch Rate Measurements

For testing the response time of the SPDT switches in the unit cell, a special back-to-back SPDT switch circuit is made, Figure 3.25, in which the antenna and MMICs are replaced with microstrip ports for connection to a microwave source and oscilloscope. The oscilloscope connected to one of the ports will record the switch turning "on" and "off". A bias tee connected between the oscilloscope and SPDT prevents the oscilloscope from loading the PIN diode bias circuitry. The optical fibers connect the SPDT switch to the optical test setup which provides complementary optical pulses with variable width.

Figure 3.24. Simulation of the time (a) and frequency (b) suppression of the switching currents generated by a 2 ns rise time/fall time 5 ns pulse period optical pulse by the HPFs. The bias switching currents on one side of the SPDT switch (blue) are filtered through HPFs using the data from Figure 3.23 with no inductors to simulates the induced switching currents on the other side of the SPDT switch (red).

Table 3.1. Measured input and output power for back-to-back SPDT switch showing maximum input power for the SPDT switch is at least 17 dBm.

Input Power (dBm)	"on" Path Output Power (dBm)	"off" Path Output Power (dBm)	Difference Difference
-5	-30.3	-65.3	-34.9
0	-25.6	-61.6	36.0
5	-20.7	-57.1	36.4
10	-15.6	-51.8	36.2
16.7	-9.1	-44.4	35.31

the SPDT switch at its fastest switch rate.

3.2.2 Maximum Input Power for SPDT Switch   The maximum microwave input power for the SPDT switch is when the PIN diodes rectify enough of the microwave signal and self-bias. When this occurs, the IL and RL will increases and the difference between the IL and ISO will decrease. The SPDT switch is tested with RF input power ranging from -5 dBm to 17 dBm while measuring the outputwave forms for the "on" and "off" paths. The output power for the two paths increases linearly with input power for the full range, while maintaining a 36 dB difference between the IL and ISO, Table 3.1.

3.3 Switch Rate Measurements

For testing the response time of the SPDT switches in the unit cell, a special back-to-back SPDT switch circuit is made, Figure 3.25, in which the antenna and MMICs are replaced with microstrip ports for connection to a microwave source and oscilloscope. The oscilloscope connected to one of the ports will record the switch turning "on" and "off". A bias tee connected between the oscilloscope and SPDT prevents the oscilloscope from loading the PIN diode bias circuitry. The optical fibers connect the SPDT switch to the optical test setup which provides complementary optical pulses with variable width.

Figure 3.24. Simulation of the time (a) and frequency (b) suppression of the switching currents generated by a 2 ns rise time/fall time 5 ns pulse period optical pulse by the HPFs. The bias switching currents on one side of the SPDT switch (blue) are filtered through HPFs using the data from Figure 3.23 with no inductors to simulates the induced switching currents on the other side of the SPDT switch (red).

71

Figure 3.25. Picture of back-to-back SPDT switch layout used to test the SPDT switches transient response. The LNA, PA and antennas have been replaced with microstrip ports for testing.

3.3.1 Optical Test Setup The optical control test setup is responsible for providing variable width optical pulses with fast rise times and fall times to the photodiodes in the back-to-back SPDT switch. The power in the optical pulses also needs to vary from less than microWatt to several milliWatts. The pulses are created by splitting the power from an Ortel 10 mW 1300 nm laser diode into two electro-optical (EO) intensity modulators. A polarization-maintaining evanescent mode coupler is used to split the optical signal from the laser while still maintaining the correct polarization orientation for the EO modulators. The Ortel laser diode is fiber pigtailed with single mode fiber iwhich scrambles the output polarization of the light from the laser. By stressing the single mode fiber, the light can be forced into one polarization mode. The EO modulators are chosen to have a 3 GHz bandwidth allowing them to generate pulses with of 0.1 ns rise and fall times.

$$\tau_{rise} = \frac{1.8}{2\pi \, BW} \qquad (3.2)$$

To achieve complete extinction of the optical pulse, the EO modulators are driven at V_π (4.5 V) by Veritech VMCM2.5MD-422 EO modulator drivers that provide the necessary gain needed to drive the 50 Ω input port of the EO modulator.

Two different setups are developed to stimulate the SPDT switch with

72
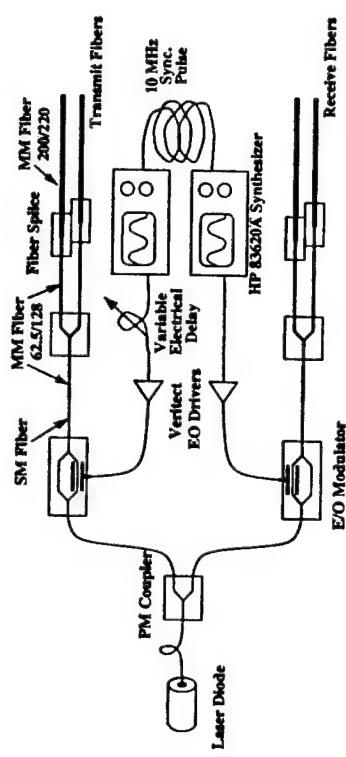
Figure 3.26. Illustration of the fast optical test setup able to generate 25 ns wide complementary optical pulses with 2 ns rise time/fall time.

optical pulses of different rise and fall times, Figure 3.26,3.27. The first setup is for high optical power, where the SPDT is expected to react in tens of nanoseconds, Figure 3.26. Two HP 83620A synthesizers are synchronized together through their 10 MHz synchronization ports. The cable connecting the two HP 83620A sweeper is long enough to delay the pulses such that the 20 MHz sine waves produced by the synthesizer are 180° out of phase. The output sine waves drive the Veritech EO modulators creating optical pulses with 25 ns pulse width and 1.5 ns rise and fall times, Figure 3.28. An additional 2 ns variable electrical delay line is used to tune
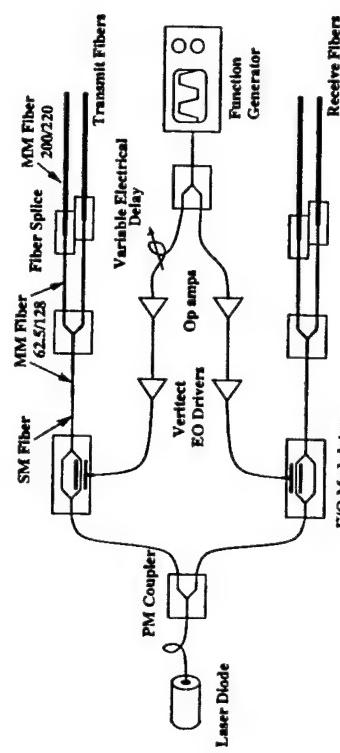
Figure 3.27. Illustration of the slow optical test setup able to generate 625 ns to 6250 ns wide complementary optical pulses with 25 ns rise time/fall time.

the setup for variations in cables and fiber lengths.

The second setup is for low optical power pulses, which requires long pulse widths, Figure 3.27. This setup uses a HP 8116A function generator to generate a square wave that in turn drives two operational-amplifier in inverting and non-inverting unity gain configurations. Again the Veritech EO drivers are used to am-plifier the operational-amplifier outputs for the EOs. This system is able to generate pulses varying in width from 625 ns to 6250 ns with a constant rise time and fall time of 25 ns, Figure 3.29.

The output of the EO modulators is split and fed to the SPDT switch. Ideally, the output fibers of the multi-mode splitter would be glued directly to the photodiodes to minimize the coupling loss as illustrated, Figures 3.30. Optical fibers have a protective jacket that prevents water from penetrating the micro-fractures in the glass which would cause the fiber to break. To glue the multi-mode fiber to the photodiodes, the outer jacket is stripped off, making the fiber extremely delicate. It proved impossible to develop a support mount for the fibers that allows access to the microwave circuits and prevents the fibers from shearing. Instead, the output fibers from the multi-mode splitter is glued to 200 μm fibers that as suspended over the photodiodes, as shown in Figure 3.31. Due to the uncertainties in the relative position of the photodiode with respect to the fiber, the fiber is move away from the photo diode so that the natural diffraction from the end of the fiber produces a 470 μm spot on the photodiode. To prevent cross-talk between fibers, baffles made from aluminum are positioned on the PDs. A 5 mil piece of FR4 is glued to the bottom surface of the baffle to prevent it from shorting out the PDs. Assuming the multi-mode fiber uniformly illuminates the spot, the loss in the coupling is 15.7 dB. Ths is too high for a practical application, and in Chapter 7 are given suggestions as to how this loss can be rduced in future designs.

(a)

(b)
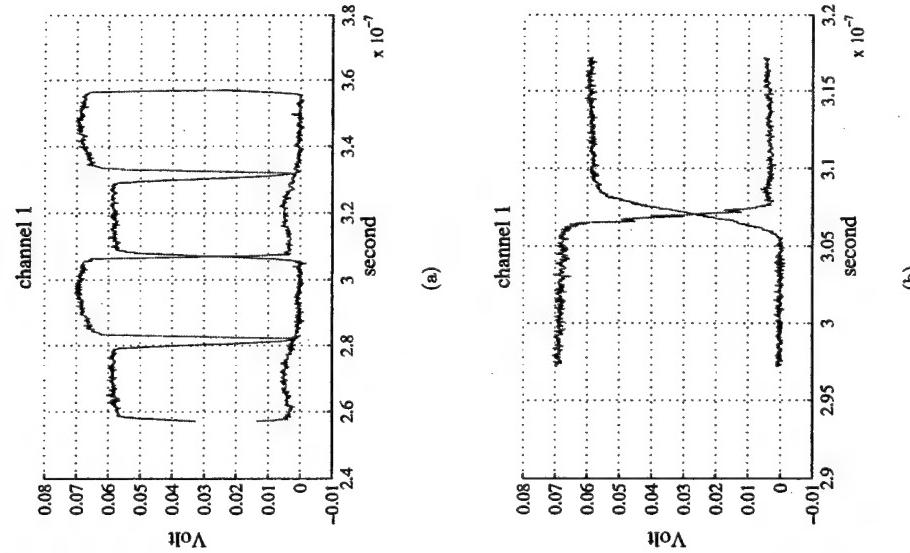
Figure 3.28. The complementary optical pulses (blue and red) for the fast optical setup (a). The rise time and fall times are shown in (b).

(a)   (b)   (c)
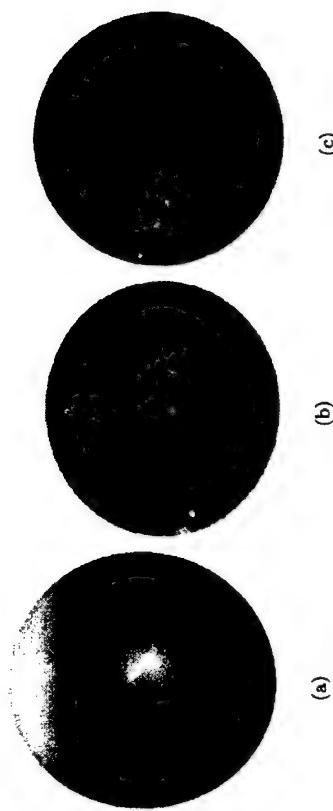
Figure 3.30. Photographs showing the 62.5 μm core to multimode fiber (a), the 80 μm active area of the photodiode (b) and the fiber and photodiode glued together (c).

Figure 3.31. Photograph of the optical mount for the back-to-back SPDT switch . The mount contains an aluminum baffle to prevent cross-talk between photodiodes.
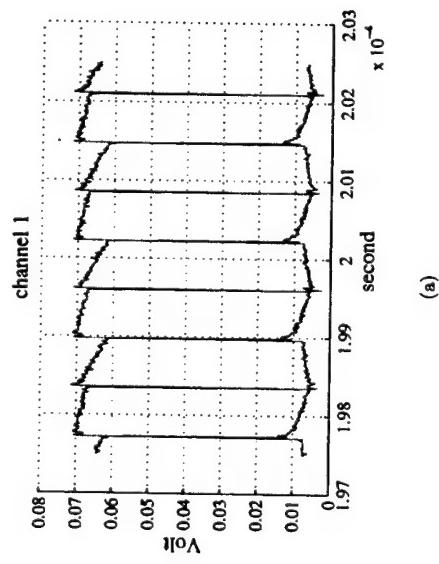
(a)
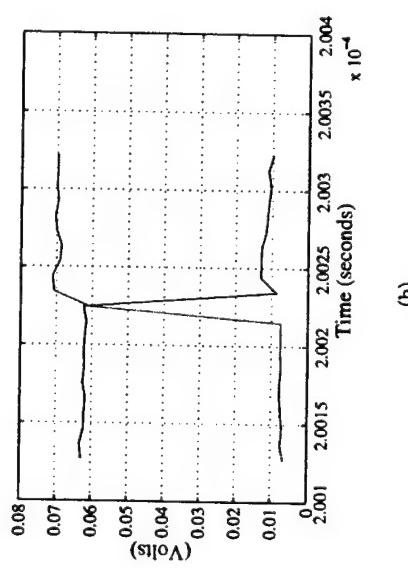
(b)

Figure 3.29. The complementary optical pulses (blue and red) for the slow optical setup (a). The rise time and fall times are shown in (b).
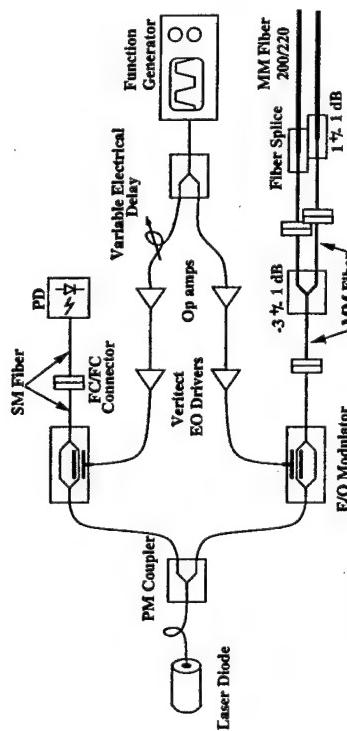
Figure 3.32. Illustration of the calibration configuration (upper arm) and the testing configuration (lower arm) of the slow optical test setup

3.3.2 Calibration of the Optical Setup

It is important to estimate the uncertainties in the amount of optical power delivered to the photodiode as a result of changes in the optical setup and mechanical variations. The two optical setups are calibrated at the output of the EO modulator with fiber pigtailed Fermionics photodiodes (upper arm in Figure 3.32). After calibration, the FC/FC optical connector is broken and the multimode fiber setup is connected (lower arm in Figure 3.32). By adding the multimode fiber setup, the calibration has an additional 4 dB of loss and 2 dB of uncertainty.

Most of the loss and uncertainty is due to the free-space coupling from the fiber to the photodiode active area. To calculate the uncertainty in the coupling, each mechanical connection between the optical mount, RF mount and substrate is described as a stochastic process. The mechanical structures are discussed in more detail in section 4.1. For example, the holes used to align the optical mount to the RF mount have an uncertainty in their position of $\pm 5.7\ \mu m$. Table 3.2 describes the probability density functions (PDFs) for the various components in the system.

The uncertainty of the photodiode placement is based on measurements of 21 photodiodes. Figure 3.33 shows the PDF for alignment of various components

Table 3.2. Properties of probability density functions used to calculated the misalignment of the photodiodes with the optical fibers.

Description	pdf	mean	variance
Hole Location Optical Mount	Uniform [-10:10]	0	33
Hole Location RF Mount	Uniform [-10:10]	0	33
Alignment Pin Bending	Uniform [-100:100]	0	3333
Fiber Location in Optical Mount Hole	Uniform [-30:30]	0	300
Photodiode Placement on Mount	Uniform [-160:160]	0	8533

in the mechanical system. The resulting uncertainty in the mechanical alignment is $\pm 93\ \mu m$, mostly due to the uncertainty in the placement of the chip photodiode on its mount. The alignment between the photodiode mount and the optical fiber has an uncertainty of $\pm 71\ \mu m$. The overall loss in the free space coupling of fiber to photodiode is 15.7 dB ± 1 dB assuming that the spot created by the optical fiber has uniform power density. The total loss and uncertainty in the optical test setup is 19.7 dB $^{+1.7\,dB}_{-2.7\,dB}$.
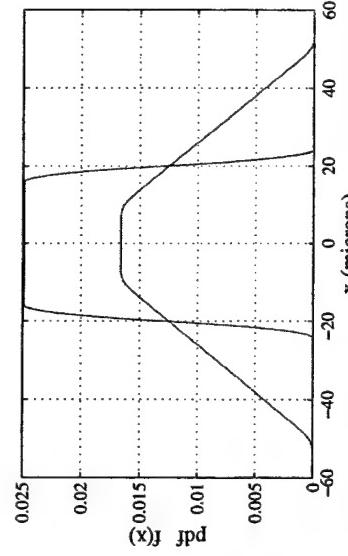
Figure 3.33. The probability density functions for the alignment for the optical fibers to the photodiode mount (blue) and the photodiodes (red).

the spot made by the optical fiber has uniform power density and not Gaussian. The minimum amount of optical power needed maintain the state of the switch is 0.25 μW.

Table 3.3: The measured rise time and fall time of the back-to-back SPDT switch.

Optical Power (μW)	Rise Time (ns)	Fall Time (ns)
15	555	469
9.3	772	785
4.95	1180	1180
2.75	2290	1600

3.3.3 Measured Switch Rate Performance

The second optical setup is used to measure the SPDT switch, since the 20 dB of optical loss limits the fastest response time for the SPDT to about 500 ns. At these slow rise times and fall times, the bias tee filters the output response, and the rise and fall times cannot be measured with the oscilloscope. The bias tee is removed and the *SPICE* model is adjusted to account for the loading of the oscilloscope. The SPDT switch is measured at four different optical powers with rise and fall times shown in Table 3.3. Figure 3.34 plots the measured rise time and fall time for the different optical powers with simulated rise times and fall times and uncertainty bounds. Interestingly enough, the SPDT switch operates better than expected. These suggests that expected loss for the optical control is less than stated earlier, probably due to the assumption that

Figure 3.34. The measured rise time (red ·) and fall time (blue ·) for the back-to-back SPDT switch. The measured performance is consistent with the simulated performance (–) and within the simulated error bounds (--).

CHAPTER 4

10 GHZ UNIT CELL AND LENS ARRAY

4.1 Unit Cell Construction

In the previous chapter the design and implementation of an optically controlled SPDT switch was presented. When this switch is inserted into an element of an active array, a number of manufacturing issues need to be solved, including electrical and mechanical ones.

The mechanical designs needs to fulfill certain requirements, while being limited to tools and materials available in the laboratory. The soft low-loss Roger's Duroid substrates used to make the microwave circuits need to be reinforced with a stiffer material to maintain their shape while being mounted into different test setups. The material used to reinforce the substrate should be a dielectric to avoid interference with the microwave circuits. If at all possible, the support structure should also protect the delicate gold bond wires on the surface of the array. The mechanical system needs to align the optical fibers to the photodiodes while allowing access to the microwave components. Precise placement of the microwave components is also critical. For convenience, all the pieces are milled on the PCB prototyping milling machine mentioned in section 2.3. The milling machine can mill FR4 (PCB material), aluminum and brass, as well as the softer Teflon-based substrates. Soft metals, such as copper gum, the milling tools, while low temperature plastics like Nylon, Delrin and polycarbonate melt from the high-speed milling tool friction.
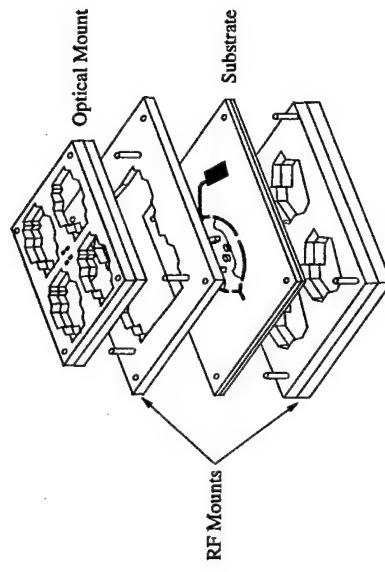
Figure 4.1. Illustration of the assembly of the FR4 mounting structures with the substrates.

Since the milling machine is limited to cutting, drilling and gouging (milling only part way through the material) on materials up to 128 mils thick, the overall unit cell is composed of a set of laminate pieces pined together, Figure 4.1. It is composed of four layers of FR4 and two layers of Duroid. The two layers of Duroid substrate contain the front and back side microwave circuits (i.e. antennas, SPDT switch, ...). The bottom three layers of the FR4 RF mount, support the soft Duroid material by sandwiching it between them. This mount also provides two sets of alignment pins used to align the substrate with the optical mount. The top two layers of FR4, referred to as the optical mount, align the optical fibers to the photodiodes while protecting the gold bond wires. Since the optical fibers are free space coupled to the photodiodes, the optical mount may be removed from the RF mount for access to the microwave circuits.

4.1.1 Unit Cell Layout   The layout of the unit cell is based on the layout of the array, such that the performance and stability of the unit cell is as close as possible to that of the array. There are two sides to the unit cell, the active component side and passive component side.

On the active component side, effort is made to have about 30 dB of isolation between any two paths that can carry the microwave signal. Since the maximum gain of the MMICs is about 16 dB, this should prevent any free space oscillations. The electrical and mechanical components are restricted to be outside an area of 20 mm by 20 mm, (section 2.5.3), around the each patch antenna. The bias lines are placed on the passive side of the unit cell to prevent oscillations due to coupling through the MMICs power connections. A tilt of 6.3° to the patch antennas allows for tighter packing of the unit cells into the array.

For good optical fiber coupling the photodiodes need to be accurately position on the substrate. The holes are cut into the active side substrate the exactly fit the ceramic mounts for a pair of photodiodes. The pressure fit of the photodiodes into these holes allows for the precise positioning. The ground plane for the active side substrate is milled away just around the holes to prevent the photodiodes from shorting. A via is provided to make the ground connection for the photodiode.

The PIN diodes and LPF capacitors are mounted with silver epoxy on 0.5 mm brass via posts in the substrate providing a good ground connection and accurate placement of the devices. In an effort to minimize the length of the bond wires to the MMICs and capacitors, they are placed on 0.38 mm (15 mil) brass platforms. Longer pass-through vias are made to connect the active and passive sides of the unit cell. The ground planes are removed from around the pass-through via holes on the both the active and passive substrates to prevent the vias from shorting to the ground plane. One set of pass through vias connects the -2.3 V bias line to the photodiodes while another set connects the 5 V bias line to the current limiting resistor and MMICs. The passive side patch antenna is also connected the active side with a pass-through via. The brass components are pictured on a penny to demonstrate their size, Figure 4.2. All of the brass pieces are made to a accuracy of 1-2 mils from brass rods or brass strips cut on the milling machine. Then the brass

Figure 4.2. Photograph showing the size of the milled brass pieces on a penny used in the construction of the unit cell.

pieces are filed to size as illustrates in Figure 4.3.

The passive side of the unit cell contains the bias line, antenna, and the lensing delay lines. Even though the MMICs have 100 pF capacitors filtering the 5 V supply voltage, additional capacitor pads are placed approximately every $\lambda/2$ along the bias-line encase bias-line oscillations occur. Since the MMICs and SPDT switch are supplied from the same bias line and their vias are placed quite close together, additional capacitor pads are place next to the switch via to suppress the switching currents from polluting the 5 V supply to the MMICs.

Besides the FR4 and substrate pieces, special aluminum plates are made to allow gold bonding on both sides of the unit cell. The hole pattern cut into them allows from maximum heating the material for gold bonding while not damaging the circuits. Figure 4.4 pictures the aluminum plate for the lens array.
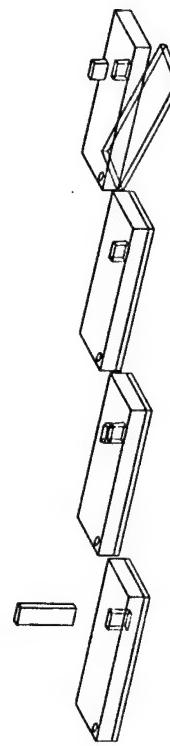
Figure 4.3. An illustration showing how to fabricate the small brass pieces used in the unit cell.

Figure 4.5. Photograph showing the active side (a) and passive side (b) of the RF mount for the unit cell with their alignment pins.

the $240\,\mu m$ optical fibers, Figure 4.6(a). The third layer contains the baffles which are 1.5 mm in diameter and 1.27 mm deep that fit over each photodiode, Figure 4.6(b). A post that rests on the substrate surface maintains the baffle height above the photodiodes. The fibers are fitted into the optical mount such that they rest 1.12 mm above the photodiodes and protrude 0.15 mm into the baffle. The optical fibers are secured in the 0.3 mm holes with quartz. The optical fibers protrude into the baffles to prevent the quartz wax from flowing over their ends in the gluing process.

4.2 Unit Cell Testing

The test setup for the unit cell is the same as the one used for testing the 8 GHz unit cell, Figure 2.36. There is 13.5 dB of loss in the unit cell due to aperture loss (4.7 dB x 2), IL (0.32 dB x 2), via loss (1 dB) and radiation efficiency (2.4 dB). The gain of the MMICs overcome the loss to give a positive gain for the unit

Figure 4.4. Photograph showing an aluminum plate that allows wire bonding on both side of the lens array.

4.1.2 RF Mount Design

The main purpose for the RF mount is to provide support for the Duroid substrate and align the substrate to the optical mount. 0.8 mm brass pins in the lower FR4 RF mount pieces match 0.8 mm holes in the substrate and 1 mm holes in the upper FR4 RF mount pieces. The 1 mm holes have tapered ends allowing easier mating of the two pieces. An additional four pins set in the second RF mount piece match four 1 mm holes in the optical mount, Figure 4.5(a). One pin fixed in the bottom RF mount piece passed through all FR4 and substrate layers to align with the optical mount. It is in the center of the photodiode cluster allowing maximum absolute alignment between the fibers and photodiodes. 20 mm by 200 mm apertures cut into the lower RF mount pieces allow the passive side antennas radiate through the RF mount, Figure 4.5(b).

4.1.3 Optical Mount Design

The optical mount is the most complicated piece made with the highest tolerances. It provides protection to the microwave circuits while maximizing the aperture (20 mm by 20 mm minimum) through which the active side patch antenna radiates. It also contains optical fibers and baffling to prevent cross-talk between the photodiodes.

The optical mount consists of three laminated layers of FR4. The top two layers contain the windows for the antennas as well as the $300\,\mu m$ holes that hold

Figure 4.7. Measured gain of the unit cell in the transmit (blue) and receive mode (red) calibrated to a "through" measurement of an aperture of $3/4\,\lambda$ by $1\,\lambda$.

30° off optical axis feeds. The half power beam widths (HPBW) for the 0° feed is 8.5° in focusing dimension and 17° in the non-focusing dimension. The HPBW for the 30° feed increases to 12.5° in the focusing dimension and 16° in the non-focusing dimension. The simulations are done using code describe in section 5.2. Due to the $1\,\lambda$ spacing, grating lobes appear at ±90° in the non-focusing dimension. The $3/4\,\lambda$ spacing causes grating lobes to appear when the main beam is steered off axis, Figure 4.9(b). Figure 4.10 shows the radiation pattern for a cut in the focusing dimension.

4.3.1 Optical Control Problems with the Array Measurements

The transmit and receive fibers from each unit cell are bundled together using quartz wax. Each bundle is polished with 10 μm to 0.5 μm polishing paper. particles size. The bundles are aligned to a 70 mW 1480 nm laser diode that is collimated using a 8 mm focal distance lens, Figure 4.11. Since the beam from the laser diode is elliptical in shape, it is impossible to get a perfectly collimated beam using a single lens. It is important to collimate the laser beam since the optical fibers are limited to an acceptance angle of 25.4° or less, determined by their numerical aperture of 0.22. The optical fiber bundles were polished by hand using a specially made jig to hold them perpendicular to the polishing surface. Unfortunately, due to human

(a)

(b)

Figure 4.6. Optical mount for the unit cell placed on the FR mount (a) showing the optical fibers and antenna apertures. The back side of the mount (b) shows the FR4 baffle and substrate resting post.

cell in a "through" measurement, Figure 4.7. The unit cell exhibits at least 30 dB of isolation in anomalous modes of operation such as: transmitting in receive mode and receiving in transmit mode. Measurement performed with the unit cell powered "off" all exhibit 30 dB of isolation demonstrating that isolation measurement is limited to edge diffraction around the unit cell.

4.3 Array Design and Testing

The lens array, Figure 4.8, is designed as a cylindrical lens with 3 row of 8 unit cells. The slant to the rows allows for tighter packing of the unit cells with $3/4\,\lambda$ spacing in the focusing dimension and $1\,\lambda$ in the non-focusing dimension. The focal distance to diameter ratio is 1 and the points of perfect focus, section 5.1, are at ±45°. Figure 4.9 shows the simulated radiation patterns for the lens with 0° and at ±45°. Unfortunately, due to human

(b)

(a)

Figure 4.9. Simulated normalized radiation patterns for the lens array with 0° off axis feed (a) and 30° off axis feed (b). The focusing dimension is the horizontal axis and the non-focusing dimension is the vertical axis.

Figure 4.10. Simulated normalized radiation patterns for the 0° off axis feed (red) and the 30° off axis feed (blue) in the focusing dimension.

(a)

(b)

Figure 4.8: The active side for the lens array (a) with optical mount (b).
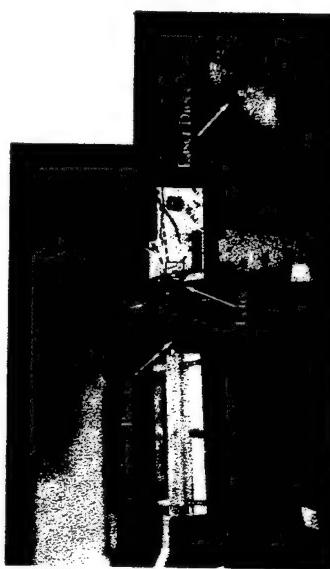
Figure 4.11. Photograph showing the coupling of the 1480 nm laser diode to the optical fiber bundle.

imperfection, the outside fibers in the bundle have rounded edges, from the wobbling of the jig.

The rounded edges and the reduced optical power at the edge of the bundle reduced the optical power coupled into the outside fibers preventing proper biasing for their unit cells. Figures 4.12 and 4.13 show the best achieved bias voltages for the PIN diodes for the array in transmit and receive modes respectively. The PIN diodes are "on" for voltages above 1 V. Voltages just below the threshold voltage present a low impedance to the switch which may not be enough to prevent oscillation.

Figure 4.12. Picture showing the oscillating and questionable unit cells for the lens in transmit mode.
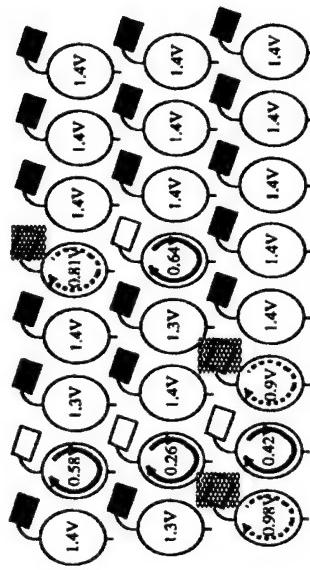
Figure 4.13. Picture showing the oscillating and questionable unit cells for the lens in receive mode.

When neither set of PIN diodes are properly biased "on", then both paths in the SPDT switch are "on" forming a ring oscillator with the LNA and PA providing the gain. With the 30 dB isolation between the unit cells, each unit cell oscillated at its own frequency, around 8 GHz and can be seen on the spectrum analyzer. By counting the number of oscillating peaks, the number of non-functioning unit cell can be determined. By moving the laser spot across the fiber bundle, it is possible to properly bias once oscillating unit cells at the expense of another unit cell oscillating. It is this test methodology that proved each unit cell is built properly and the optical control is the reason for the array oscillation.

4.3.2 Array Pattern Measurements   At the time of the development of the optically controlled lens array, an anechoic chamber was not available with a rotating stage that could accommodate the optical feed setup. Therefore preliminary radiation pattern measurements are taken with a rotating stage mounted to an optical table, Figure 4.14. A horn antenna mounted on a rotating arm behind the lens array provides the feed for the array. Patterns are taken for the feed at 0° and 30° off optical axis. The 30° feed radiation pattern does not steer the main beam through the fiber bundles shown in, Figure 4.14. A spectrum analyzer is used to measure the transmit and receive power while filtering out the radiated power
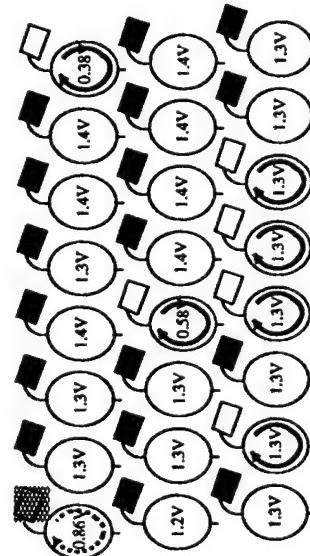
(a)

(b)

Figure 4.15. The measure normalized radiation patterns in the focusing dimension for the 0° off axis (a) and the 30° off axis (b) feeds. The transmit patterns are in (blue) and the receive patterns are in (red).

Figure 4.14. Photograph showing the setup used to measure the radiation patterns for the lens array.

from the oscillating unit cells.

Unfortunately, in this environment, multi-path reflections and power leaking around the lens array to the feed limits the accuracy of the pattern measurements. The effects of multi-path are estimated by monitoring the spectrum analyzer power levels as objects and people move about in the room. After looking at the pattern measurements of the active and passive arrays, the best estimate for the accuracy in the radiation patterns is about 10 dB to 15 dB down from the main lobe for the active array. No discernible array pattern can be deduced from the passive arrays, implying that the errors due to multi-path and power leakage are on the order of the gain of the lens. Radiation patterns can be discern for the active lens array due to the gain of the LNA and PA.

Radiation patterns taken in the focusing dimension, show an increase in side lobe power and an asymmetry in the main beam, Figures 4.15. The characteristic grating lobes do appear in the pattern for the 30° scan angle, Figure 4.15(b). Patterns simulations for the transmit and receive cases with the appropriate oscillating unit cells turned off and a patch antenna feed, show an increase in side lobe and some asymmetry, Figure 4.16. Most of the asymmetry in the radiation patterns must be due to multi-path errors. Three-dimensional pattern simulations show a reduction in the radiation pattern structure due to the "off" unit cells, Figures 4.17, 4.18.

(b)

(a)
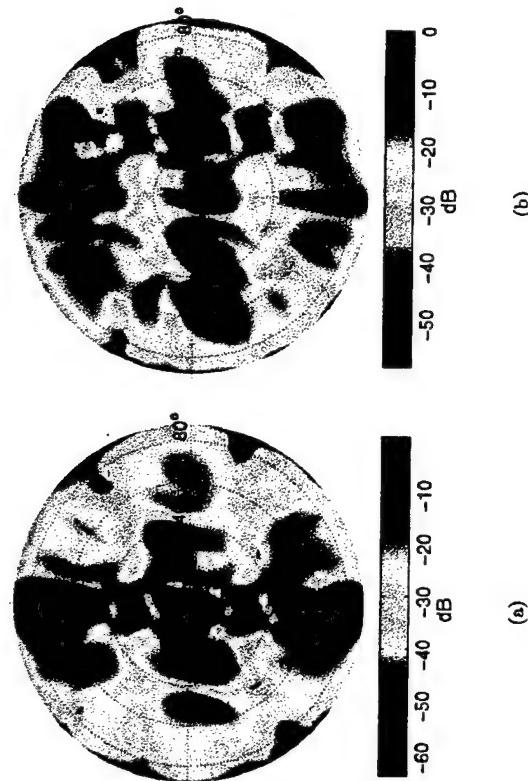
Figure 4.17. The simulated normalized radiation patterns for the 0° off axis (a) and the 30° off axis (b) feeds for the lens array in receive mode.

To estimate the effects of the fiber bundles on the radiation pattern, measurements are done with the passive array by itself, with the FR4 mount and with the FR4 mount and optical fibers. A different test setup is used for the passive array where the array held fix and the source horn moves. This setup seems less prone to multi-path errors and the patterns are taken in the evening when the laboratory is empty. A major source of error in these measurements is maintaining the horn pointing direction to the array. The pattern measurements for the array with the main beam steered through the fiber bundle shows about 2 dB reduction in power for the FR4 mount and the FR4 mount with optical fibers, Figure 4.19. From section 2.5.3 the FR4 mount does attenuate the patch radiation pattern off axis. There is not a noticeable difference between the FR4 and the FR4 mount with optical fibers, suggesting that the FR4 mount dominates the reduction in the radiation pattern.

(a)

(b)

Figure 4.16. The simulated normalized radiation patterns for the 0° off axis (a) and the 30° off axis (b) feeds in the focusing dimension. The transmit patterns are in (blue) and the receive patterns are in (red) with the oscillating unit cells turned off. The radiation patterns for a fully function lens array are in (green).
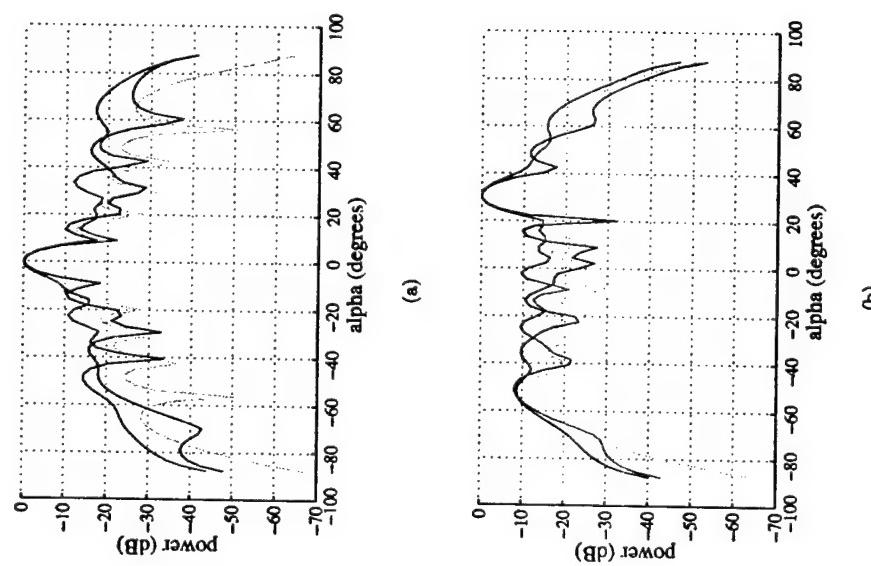
4.4 Shaped Pulse Control of SPDT Switch

For controlling very large antenna arrays, the optical control pulses maybe further optimized in shape to provide more efficient use of the optical power. As mentioned in section 3.1.4, the controlling factor on how fast the SPDT switch changes states is the RC time constant formed from the gate capacitance of the MESFET and the "on" resistance of the photodiode. The "on" resistance of the photodiode is a model for the generated carriers in the photodiode discharging the stored charge on the gate capacitance of the MESFET. The more carriers there are, the faster the capacitor discharges. Once the capacitor is discharged, there is not need for the large numbers of carriers and the optical power can be reduced to a level where the gate of the MESFET is held at the proper voltage potential. From the measurements in section 3.3.3, this requires only $0.25\,\mu W$ of optical power.

The low "background level" power level suggests a novel control methodology for the SPDT switch, where the photodiodes are pulsed at a high optical power long enough to allow the switch to change state and then reduced to the minimum power for the rest of the period. For the switch rates measured in section 3.3.3, the required optical energy needed to change the state of the switch stays reasonably constant with switch rate at about $10\,pJ$ to $20\,pJ$, which is far less than the reported energy required to switch a MEMS switch [29]. Figure 4.20 shows a *SPICE* simulation for this control scheme where the complementary optical pulses (blue and red) are switched from $7\,mW$ of optical power to $0.25\,\mu W$. The bias voltage across the two sets of PIN diodes (purple and yellow) switch for "on" to "off" and vise versa, showing the switch changes state.

The real motivation for this type of control comes into play when large arrays are being controlled. For example, assuming a commercially available $100\,mW$ $1480\,nm$ laser diode used in yttrium doped fiber amplifiers is used to control an array at its fastest rate of $2.4\,ns$ rise time for periods of $10\,ns$. How large of an array can

(a)

(b)
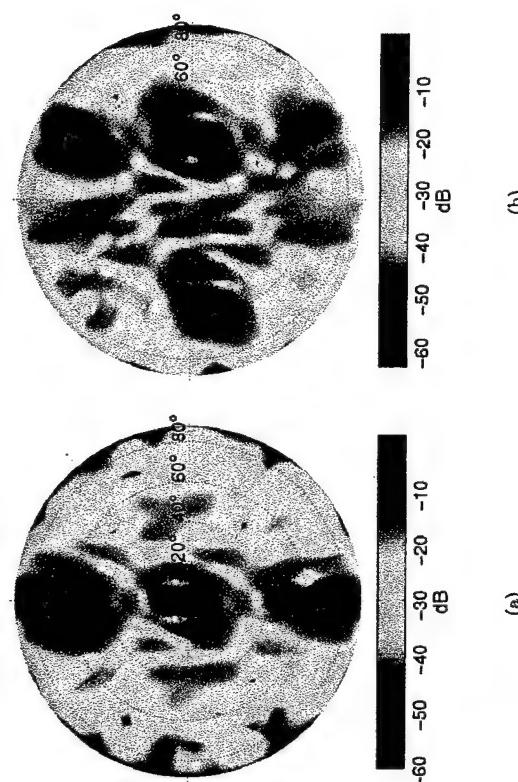
Figure 4.18. The simulated normalized radiation patterns for the $0°$ off axis (a) and the $30°$ off axis (b) feeds for the lens array in transmit mode.
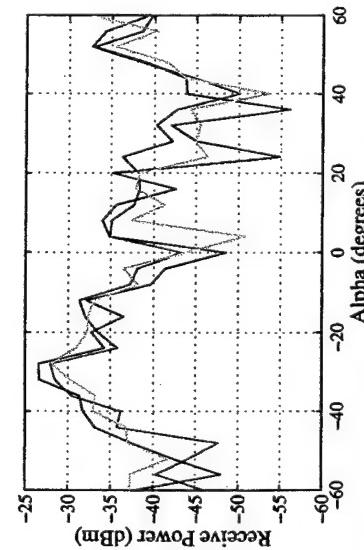
Figure 4.19. Measured passive array radiation patterns for the array (blue), array with FR4 mount (red), and array with FR4 mount and optical fibers. The $30°$ off axis feed steers the main beam through the fiber bundles.
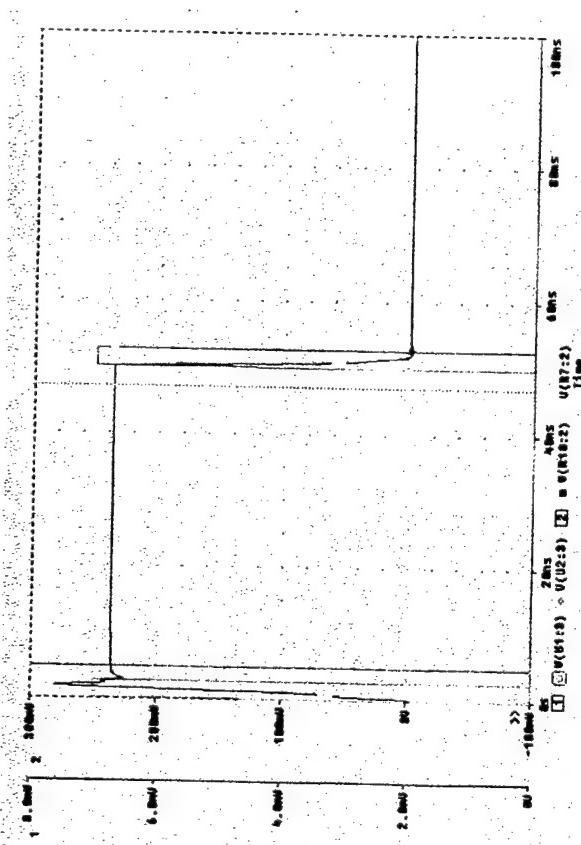
Figure 4.20. *SPICE* simulated performance for the unit cell using shaped optical pulses to reduces the required optical power needed for fast switching. The voltage sources representing the optical signals are (blue and red, far left y-axis scale) and the bias voltages for the two sets of PIN diodes are in (purple and yellow, right y-axis scale)

be controlled with and without the shaped optical pulses? The 100 mW limit on the laser diode is the average power limit dictated by the laser ability to dissipate heat. For a 2.4 ns rise time, 9 mW of optical power is needed per diode or 18 mW per unit cell. For the non-shape optical pulse that allows a

$$\frac{100\,\text{mW}}{18\,\text{mW}} = 5.5 \qquad (4.1)$$

unit cells to be controlled. For the shape optical pulse the average optical power needed per unit cell is

$$2\frac{2.4\,\text{ns}}{10\,\text{ns}}9\,\text{mW} + 2\frac{7.6\,\text{ns}}{10\,\text{ns}}0.25\,\mu\text{W} = 4.3\,\text{mW} \qquad (4.2)$$

allowing

$$\frac{100\,\text{mW}}{4.3\,\text{mW}} = 23.2 \qquad (4.3)$$

unit cells to be controlled. As the pulse period increases, the power savings for the shaped pulse become more significant.

CHAPTER 5

LENS ARRAYS DESIGN

5.1 Constrained Lens Array Theory

As presented in chapter 1, adaptive array communications systems use a complex weighted sum of the array element outputs to steer a beam in the direction of the desired user and creating nulls in the directions of any unwanted users. In "standard" planar arrays, the information about the direction of the desired user is contained in the phase variations across the array, thus the adaptive algorithm uses all the elements in the array to steer the beam towards the user. However, lens arrays focus signals from different directions onto a focal surface, where the resulting image spatial decorrelates the different signal allowing for more efficient processing of the information. This is equivalent to transforming the problem from a phase-space representation (planar array) to a beam-space representation (lens array). The focusing quality of the lens array affects the overall performance of the adaptive lens array system. In order to understand how lens arrays affect the performance of an adaptive array system, it is important to understand how the design of a lens array affects its ability to focus.

There are several different types of lens arrays that date back to the 1940's, in which stacked metal plates form a waveguide lens used to enhance early RADAR performance [47]. Gent introduced "the Boot Lace Aerial" in 1957 in which transmission lines connect two antenna arrays to form a lens [48]. The two antenna arrays

Figure 5.1. "The Boot Lace" lens array showing the four degrees of freedom: 1. Curved non-feed surface; 2. Curved feed surface; 3. Non-feed and feed elements relative offset; and 4. Delay.

function as the front and back surfaces of the lens, and the transmission lines introduce the delays through the lens. The surface of the lens on the side of the focal surface will be referred to as the feed side, and the other side the non-feed side. The transmission lines are longer in the center, in analogy to an optical lens being thicker in the center. By using transmission lines instead of waveguides structures, the lens is non-dispersive over a very large bandwidth.

In principle, the lens arrays can be categorized by the numbers of degrees of freedom in their design. There are up to four degrees of freedom in lens arrays, as indicated Figure 5.1:

1. Non-feed side curved surface;
2. Feed side curved surface;
3. The relative positions between radiating pairs of elements in the two arrays; and
4. The electrical delay between elements.

The number of degrees of freedom in the design determines the number of perfect focal points on the focal surface, one for each degree of freedom. Roa [49] and Rappaport and Zaghloul [50] have reported designs with four perfect focal points. Roa placed the focal points on a straight line where Rappaport and Zaghloul placed them

At all other points on the focal surface, the path delay error follows closely a cubic function, Figure 5.3(a), suggesting "coma" aberrations [24] which can be reduced by re-focusing. McGrath found that a new focal surface ($G(\theta)$) given by equation (5.3) minimizes the RMS error in path length integrated over the aperture of the lens, Figure 5.3(b). The $G(\theta)$ is given by:

$$G(\theta) = F \sec(\theta_0)\left[1 + \frac{1}{2}\frac{\sin^2\alpha\sin^2\theta}{(1 - \sec\alpha)(1 + \sin\alpha\sin\theta)}\right]$$ (5.3)

$$\alpha = \sin^{-1}\left(\frac{r_{max}}{F}\right)$$ (5.4)

where r_{max} is the radius of the lens. Equations (5.1, 5.2, and 5.3) are given in terms of radial coordinates of the elements, therefore a three-dimensional array can be designed by imposing symmetry about the optical axis. Roa states that in practical situations, this axial symmetry is not possible, and a lens has a cone of best focus instead of a cone of perfect focus at θ_0. Figure 5.3 also shows how the cone of best focus and the re-focused focal surface try to minimize the average error by letting the phase be both positive and negative.

Since the planar constrained lens only has two degrees of freedom, it can be uniquely specified by two design parameters (F/D and θ_0) and the feed side element spacing, where D is the diameter of the aperture. Figure 5.4 shows the layout of two different lens arrays that are 4λ in diameter and have inter-element spacings of $\frac{\lambda}{2}$. The first lens design has F/D=0.6 and $\theta_0 = 0$, and the second design has F/D=0.6 and $\theta_0 = 45°$. The red x's and yellow o's represent the centers of the non-feed side and feed side radiating elements, respectively, with the focal surface for the lenses shown in blue. The first thing to note is that for small enough F/D and θ_0, it is possible to have a lens design that is unrealistic since the feed side elements lie outside the array aperture. By increasing θ_0, the elements are brought in but at the cost of deforming the focal surface away from the lens which results in increased loss. As in planar array design, the inter-element space in the lens array needs to be $\frac{\lambda}{2}$

Figure 5.2. A McGrath planar constrained lens with two degrees of freedom: 1. Non-feed and feed elements relative offset and 2. Delay.

on a circle, symmetric about the optical axis. Rotman and Turner [51] presented a three-focal point array lens where the front non-feed side is constrained to be flat. Two of the three focal points lie symmetric about the optical axis, and the third is on the optical axis.

The lens array presented first by McGrath [2] is of most interest to this project since it constrains both the non-feed side and the feed side to be flat, Figure 5.2. With only two degrees of freedom, the two focal points lie symmetrically around the optical axis at angles $\pm\theta_0$. The equations that determine the location and the delay for each element in the array are derived from Figure 5.2, such that the path delays for a plane wave incident at an angle θ_0 are the same for any element pair in the array. Equation (5.1) relates the location of a feed-side element (ρ) in terms of the non-feed side elements (r), focal distance (F) and angle of perfect focus (θ_0).

$$\rho = \sqrt{\frac{F^2 - r^2\sin^2\theta_0}{F^2 - r^2}}$$ (5.1)

The delay for each element (W) is determined by the non-feed side element position, focal distance and the unit delay for the center element (W_0):

$$W = F + W_0 - \frac{1}{2}\sqrt{F^2 + \rho^2 - 2\rho F\sin\theta_0} - \frac{1}{2}\sqrt{F^2 + \rho^2 + 2\rho F\sin\theta_0}$$ (5.2)

Figure 5.4. Example of two lens designs with F/D=0.6 and cone of best focus, $\theta_0 = 0^\circ$ (a) and $\theta_0 = 45^\circ$ (b). Red x's depict the centers of the non-feed side antennas and yellow o's those of the feed side antennas. The focal surface of the lenses are shown in blue.

by $\frac{\lambda}{2}$ to prevent grating lobes when steering off axis, [21]. However, the best way to pack the elements (unit cells) into the lens aperture is yet to be determined.

5.2 Lens Array Modeling

a lens arrays, only approximates a real microwave lens. The finite sampling of the aperture and the approximation of the refraction through a lens with delay lines introduce loss and aberrations in the image of the lens. As from phase errors, there are several other factors in the physics of lens arrays that also lend to the loss and aberrations in images. For example, the polarization and radiation patterns of the antenna elements in the lens affect how the re-radiated signals are focused on the image for different scan angles. The path loss difference between the different

Figure 5.3. Normalized phase error versus normalized aperture coordinates for a spherical focal surface (a), and a refocused focal surface (b). (Scanned for [2]

elements in the lens and a point on the focal surface affects how the different signals interfere.

An analytical solution for the aberrations and loss in a lens array system is, however the physics of the problem makes a analytical solution intractable. Instead, a numerical model for the lens array is developed to characterize the aberration and loss. The model uses radiation coupling equations based on Fraunhoffer diffraction theory, since the physical size of the lens system is beyond the resources of most full-wave simulators. Using radiation coupling equations, the model will not be able to predict high order interactions between elements in the array and elements on the focal surface. The model should be flexible enough to allow changes in lens design, different types of sources excitations and focal surface detectors. It also needs to be able to simulate various microwave environments (i.e. multi-path, multi-user, moving sources, changing polarization, etc.) so the effects of lens arrays in adaptive antenna systems may be explored.

5.2.1 Physical Bases for the Numerical Model

In the model for the lens array, only far-field (Fraunhoffer) radiation coupling is taken into account [45, 52]. This coupling is the result of spherical electromagnetic waves propagating from one antenna to another. The effective length is used to describe the coupling between antennas since it preserves both the gain and polarization of the antenna. The electric field at a point in the far field induced by an antenna with input current I_{in} is given by

$$\vec{E} = -j\eta \frac{kI_{in}}{4\pi R}\overrightarrow{l_{eff}}(\theta,\phi)e^{-jkR} \qquad (5.5)$$

where $\overrightarrow{l_{eff}}$ is a vector describing the polarization and gain of the electric field for an antenna in the direction (θ, ϕ). k is the propagation constant for a plane wave, and R is the distance between the center of the antenna and the point of observation. The induced open circuit voltage, V_{oc}, at the antenna port with an incident electric
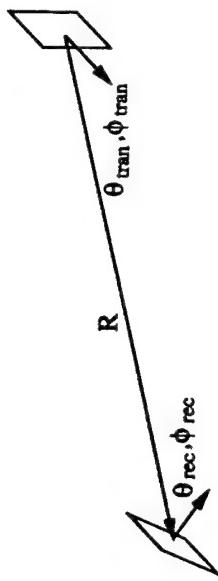
Figure 5.5: Coordinate relationships for calculating the coupling of two antennas.

field $\vec{E}$ is given by:

$$V_{oc} = \vec{E}^{*} \cdot \overrightarrow{l_{eff}}(\theta,\phi) \qquad (5.6)$$

A^{*} denotes conjugate transpose of the electric field vector. The coupling between a transmit and receive antenna that are separated by, R, and in the directions $(\theta_{tran}, \phi_{tran})$ and $(\theta_{rec}, \phi_{rec})$ for transmit and receive angles, respectively, as shown in Figure 5.5, is given by:

$$V_{oc} = -j\eta\frac{kI_{in}}{4\pi R}\overrightarrow{l_{eff}}^{*}(\theta_{rec}, \phi_{rec}) \cdot \overrightarrow{l_{eff}}(\theta_{tran}, \phi_{tran})e^{-jkR} \qquad (5.7)$$

$\overrightarrow{l_{eff}}$ for both antennas are in a common coordinate system for the dot product of the effective lengths to accurately describe the polarization interaction. This can be rewritten as:

$$V_{oc} = j\eta\frac{kI_{in}}{4\pi R}\left|\overrightarrow{l_{eff}}(\theta_{rec}, \phi_{rec})\right|\left|\overrightarrow{l_{eff}}(\theta_{tran}, \phi_{tran})\right| \\ \left(\overrightarrow{\rho_{eff}^{*}}(\theta_{rec}, \phi_{rec}) \cdot \overrightarrow{\rho_{eff}}(\theta_{tran}, \phi_{tran})\right)e^{-jkR} \qquad (5.8)$$

Equation (5.8), explicitly separates the antenna gains $|l_{eff}|$ and polarization $\overrightarrow{\rho}(\theta, \phi)$ in to a form more useful for the model. All calculations are done at a single frequency with narrowband systems in mind.

Using Equations (5.5, 5.6, and 5.8) all the electric fields on the lens array and focal surface can be calculated. For example, assume we have a source antenna in the near field of the lens array and a far-field plane wave incident on the lens array.

Figure 5.7. The coupling between two antennas, each with its own local coordinate systems

referring to Figure 5.7, Equation (5.8) can be rewritten in a form as

$$V_{oc} = j\eta \frac{k I_{in}}{4\pi R} \left| \overrightarrow{l_{eff}}(\theta_{rec}, \phi_{rec}) \right| \left| \overrightarrow{l_{eff}}(\theta_{tran}, \phi_{tran}) \right|$$

$$\overrightarrow{\rho^*_{eff}}(\theta_{rec}, \phi_{rec}) \cdot [J_{rec,sphere}(\theta_{rec}, \phi_{rec})] \cdot [J^*_{rec,globe}] \cdot$$

$$[J_{tran,globe}] \cdot [J^*_{tran,sphere}(\theta_{tran}, \phi_{tran})] \cdot \overrightarrow{\rho_{eff}}(\theta_{tran}, \phi_{tran}) e^{-jkR} \quad (5.9)$$

$[J_{rec,globe}]$ transform local receive Cartesian coordinates to global receive Cartesian coordinates. $[J_{rec,sphere}(\theta_{rec}, \phi_{rec})]$ transforms receive Cartesian coordinates to receive spherical coordinates in a direction $(\theta_{rec}, \phi_{rec})$. Since Jacobian matrices are unitary, their inverse is the conjugate transpose represented by $[J^*]$ and performs the inverse coordinate transformation, [53].

Using a circuit model to describe the connection of non-feed side elements to feed side elements, Figure 5.8, the input current for the feed side antenna can be calculated from the induced open circuit voltage at the non-feed side antenna as:

$$I_{in,NFD} = \frac{V_{oc} G_{ain}}{R_{FD} + R_{NFD}} \quad (5.10)$$

where R_{FD} and R_{NFD} are the feed side and non-feed side antennas resistances, respectively. Here the amplifier gain is a current gain and the antenna impedances are assume to be real and matched to the amplifiers.

By cascading Equation (5.9) for a near-field source or Equation (5.6) for a far-field planes with Equation (5.10) for propagating through the lens array and

Figure 5.6. Example of lens array system with far-field source (magenta hexagon), near-field source (magenta circle) and a detectors on the the focal surface (green squares).

The lens array itself is populated with antenna elements on both sides, each having their own orientation, connected through delay lines and amplifiers. The antenna elements on the feed side of the array radiate to antennas placed on the focal surface, or other points, behind the array. Figure 5.6 illustrates the lens system with the near field source represented by a magenta circle and the far-field source represented by a magenta hexagon in the direction of arrival 6λ from the center of the lens in the direction of reception. The non-feed side antennas are represented by red x's and feed side antennas represented by yellow o's at their positions in the array. The detector antenna (green) is on the focal surface (blue). Each source, array element, and detector has its own orientation represented by its own local coordinate system.

Using Jacobean matrices, [J], to perform coordinate transformations from local to global coordinate systems and from spherical to Cartesian coordinate systems

Figure 5.8. Circuit model for the non-feed side and feed side antennas in a lens array.

Equation (5.9) for feed side elements radiating to a detector, the open circuit voltage at the detector can be determined. From Equation (5.10), lens arrays fundamentally have 3 dB more loss than an equivalent planar array due to the conversion of radiated power to voltage and back to radiated power prior to detection (reception).

5.2.2 Antenna Element Models   The equations presented in section 5.2.1 describe the radiation in terms of the antenna effective lengths. One of the requirements for a numerical model for a lens array, is the ability to describe arbitrary antennas. A standard way to describe an antenna's effective length is through an integral of its surface currents [45, 52]. Depending on the antenna, this could be an electric current (e.g. in a dipole) or a magnetic current (e.g. in a cavity model for a patch antenna). in a lossless antenna, the total radiated power is equal the total input power (P_{in}) at the feed. This is sometimes written as a relationship between input resistance, R_{ant}, and radiation resistance, R_{rad} [45]:

$$P_{rad} = |I_0|^2 R_r \tag{5.11}$$

$$P_{in} = |I_{in}|^2 R_a \tag{5.12}$$

$$\frac{R_a}{R_r} = \frac{|I_0|^2}{|I_{in}|^2} \tag{5.13}$$

where the I_0 and I_{in} are the electric current density for the antenna and input current at the feed of the antenna respectively.

The radiation resistance is essentially a normalization of the total power

radiated:

$$R_{rad} = \frac{1}{|I_{in}|^2} \iint_S \vec{E} \times \vec{H}^* \cdot dS$$
$$= \frac{1}{\eta |I_{in}|^2} \iint_S |\vec{E}|^2 \cdot dS \tag{5.14}$$

where η is the free space impedance, 377Ω.

If the model has to perform a power normalization integral each time the effective length of the antenna is needed, it will run prohibitively slow. Fortunately, many antennas like dipoles, microstrip patch antennas and slot antennas have a structure to their electric and magnetic surface currents that determine their radiation patterns. For dipole antennas, all the calculations including the normalization constant can be done ahead of time and stored in a file, however for patch and slot antennas whose radiation pattern changes for different shape antennas, this is not true. For patch and slot antennas it is possible to calculate the general shape of the radiation pattern as a function of the patch shape apriori and then perform the surface integral once to calculate the normalization constant.

It is possible to relate the integral of the radiated power to an integral of the prototypes effective length. Then by calculating the normalization constant from the integral of the prototype's effective length a single set of equations can be used for the power normalization and the radiation coupling calculation. The total radiated power by an antenna is given by

$$P_{rad} = \iint_S \vec{E} \times \vec{H}^* \cdot dS$$
$$= \frac{1}{\eta} \iint_S |\vec{E}|^2 \, dS \tag{5.15}$$

The electric field is calculated from the electric current density vector, $\vec{J}$, as

$$\vec{E_J} = \frac{-k\eta}{4\pi R}e^{-jkR}\iint_S \vec{J}e^{-jkR\cos\psi}\,dS \quad (5.16)$$

$$= \frac{-k\eta}{4\pi R}e^{-jkR}\vec{N}$$

$$\vec{N} = \iint_S \vec{J}e^{-jkR\cos\psi}\,dS \quad (5.17)$$

where ψ is the angle between the vectors that point to the current element and the observation point in space. For a magnetic current density vector, $\vec{M}$, the electric field is

$$\vec{E_M} = \frac{k}{4\pi R}e^{-jkR}\hat{a}_r \times \iint_S \vec{M}e^{-jkR\cos\psi}\,dS \quad (5.18)$$

$$= \frac{-k}{4\pi R}e^{-jkR}\hat{a}_r \times \vec{L},$$

$$\vec{L} = \iint_S \vec{M}e^{-jkR\cos\psi}\,dS, \quad (5.19)$$

where $\hat{a}_r$ is the unit vector for the direction of observation. Combining Equation (5.5) with equations (5.16), (5.17), (5.18), and 5.19) gives the effective length in terms of electric and magnetic currents:

$$\overrightarrow{L_{eff,J}}(\theta,\phi) = \frac{\vec{N}}{I_{in}} = \frac{J_0\,\overrightarrow{N'}(\theta,\phi)}{I_{in}} \quad (5.20)$$

$$\overrightarrow{L_{eff,M}}(\theta,\phi) = \frac{-\hat{a}_r \times \vec{L}}{\eta I_{in}} = \frac{-\hat{a}_r M_0 \times \overrightarrow{L'}(\theta,\phi)}{\eta I_{in}} \quad (5.21)$$

$N'(\theta,\phi)$ and $L'(\theta,\phi)$ represent the surface integral for the antennas per Equations (5.17 and 5.19) and J_0 and M_0 are normalization constants for power conservation. If both electric and magnetic currents are present then the effective length is give by

$$\overrightarrow{L_{eff,JM}} = \overrightarrow{L_{eff,J}} + \overrightarrow{L_{eff,M}} \quad (5.22)$$

Combining equation (5.15) with (5.16) results in an equation for the input antenna

resistance, as follows:

$$P_{rad} = \frac{J_0^2}{2\eta}\iint_S \left|\frac{-jk\eta}{4\pi R}e^{-jkR}\overrightarrow{N'}\right|^2 dS \quad (5.23)$$

$$\frac{|I_{in}|^2}{2}R_{ant} = \frac{J_0^2 k^2\eta}{32\pi^2 R^2}\iint_S \left|\overrightarrow{N'}\right|^2 dS \quad (5.24)$$

$$R_{ant} = \frac{J_0^2 k^2\eta}{16\pi^2 R^2}\iint_S \left|\frac{\overrightarrow{N'}}{I_{in}}\right|^2 dS \quad (5.25)$$

Note that Equation (5.25) contains the effective length in the integral. Now it is possible to solve for the normalization constant J_0 in terms of the antenna input resistance, and the input current. For the lens model both the input resistance and input current are taken to be unity for unit input power in the calculation of the normalization constant, J_0:

$$\left|\frac{1}{J_0}\right|^2 = \frac{k^2\eta}{16\pi^2 R^2}\iint_S \left|\overrightarrow{N'}\right|^2 dS \quad (5.27)$$

A similar set of equation can be written for magnetic currents:

$$P_{rad} = \frac{M_0^2}{2\eta}\iint_S \left|\frac{jk}{4\pi R}e^{-jkR}\hat{a}_r \times \overrightarrow{L'}\right|^2 dS \quad (5.28)$$

$$\frac{|I_{in}|^2}{2}R_{ant} = \frac{M_0^2 k^2}{32\eta\pi^2 R^2}\iint_S \left|\hat{a}_r \times \overrightarrow{L'}\right|^2 dS \quad (5.29)$$

$$R_{ant} = \frac{M_0^2 k^2\eta}{16\pi^2 R^2}\iint_S \left|\frac{-\hat{a}_r \times \overrightarrow{L'}}{\eta I_{in}}\right|^2 dS \quad (5.30)$$

$$\left|\frac{1}{M_0}\right|^2 = \frac{k^2}{16\eta\pi^2 R^2}\iint_S \left|\hat{a}_r \times \overrightarrow{L'}\right|^2 dS \quad (5.31)$$

When both electric and magnetic currents are present and their scaling constants are related by $J_0 = CM_0$, then the new normalization equation is given by

$$\left|\frac{1}{J_0}\right|^2 = \frac{k^2\eta}{16\pi^2 R^2}\iint_S \left|\overrightarrow{N'} - C\frac{\hat{a}_r \times \overrightarrow{L'}}{\eta}\right|^2 dS \quad (5.32)$$

(a)

(b)
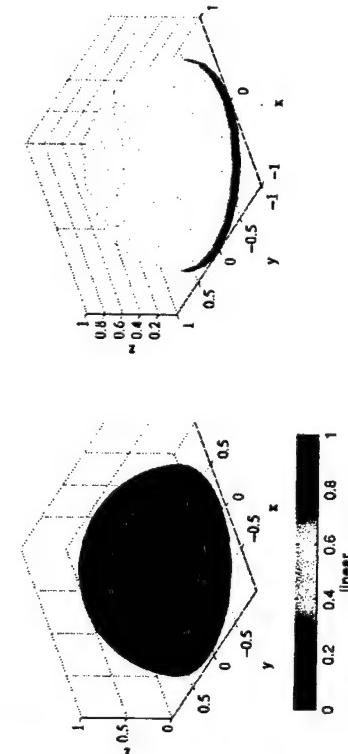
Figure 5.9. The radiation pattern (a) and far-field electric field pattern (b) for a semi-directional antenna. The electrical field vectors are plotted tangent to a unit sphere with their tails denoted as circles.

There are two principle antennas used in the model simulations, semi-directional and patch antenna. A semi-directional antenna radiates equally well in all directions in a hemisphere and nowhere else, Figure 5.9(a). The electric field polarization is chosen to be consistent with a $\hat{a}_y$ oriented electric surface currents density vectors, Figure 5.9(b). When Equation (5.27) is applied to the semi-directional, antenna the normalization constant becomes $J_0 = \sqrt{\frac{2}{\eta\pi}}$

To calculate the normalization constant for the patch antenna, the cavity model is used [45], Figure 5.10. The cavity model represents the radiation from fringing electric fields off the patch as two magnetic currents. To be consistent with the semi-directional antenna, the magnetic currents are chosen to be in the $\hat{a}_x$ direction, $\overrightarrow{M_x}$, so that the radiated electric field is polarized along $\hat{a}_z$. By using the cavity model for the patch antenna, the numerical model only needs the electric

Figure 5.10: Cavity model for patch antenna.

length (L_e), width (W_e) and height (h_e) to calculate the normalization constant and the all subsequent effective lengths. The relevant equations are given below.

$$l_{eff,\theta} = \frac{-L_\phi}{\eta I_{in}} \tag{5.33}$$

$$l_{eff,\phi} = \frac{L_\theta}{\eta I_{in}} \tag{5.34}$$

$$L_\theta = 2|\overrightarrow{M_x}|h_e W_e \cos\theta \cos\phi \, \text{sinc}(kh_e\cos\theta) \tag{5.35}$$
$$\text{sinc}\left(k\frac{W_e}{2}\sin\theta\cos\phi\right)\cos\left(k\frac{L_e}{2}\sin\theta\sin\phi\right)$$

$$L_\phi = -2|\overrightarrow{M_x}|h_e W_e \sin\phi \, \text{sinc}(kh_e\cos\theta) \tag{5.36}$$
$$\text{sinc}\left(k\frac{W_e}{2}\sin\theta\cos\phi\right)\cos\left(k\frac{L_e}{2}\sin\theta\sin\phi\right)$$

a similar set of equations are derived for $\hat{a}_y$ oriented currents, $\overrightarrow{M_y}$, the equations are,

$$L_\theta = 2|\overrightarrow{M_y}|h_e L_e \cos\theta \sin\phi \, \text{sinc}(kh_e\cos\theta) \tag{5.37}$$
$$\text{sinc}\left(k\frac{L_e}{2}\sin\theta\sin\phi\right)\cos\left(k\frac{W_e}{2}\sin\theta\cos\phi\right)$$

$$L_\phi = 2|\overrightarrow{M_y}|h_e L_e \cos\phi \, \text{sinc}(kh_e\cos\theta) \tag{5.38}$$
$$\text{sinc}\left(k\frac{W_e}{2}\sin\theta\sin\phi\right)\cos\left(k\frac{L_e}{2}\sin\theta\cos\phi\right)$$

With equations (5.33) to (5.38) the model can simulate any polarize patch antenna including elliptical polarization by rotating $\overrightarrow{M_y} = C\overrightarrow{M_x}$, where C is a complex constant. The normalization constants, M_0, needs to be calculated only once and them stored in memory for future reference. For example, the patch antenna used in

(a)

(b)
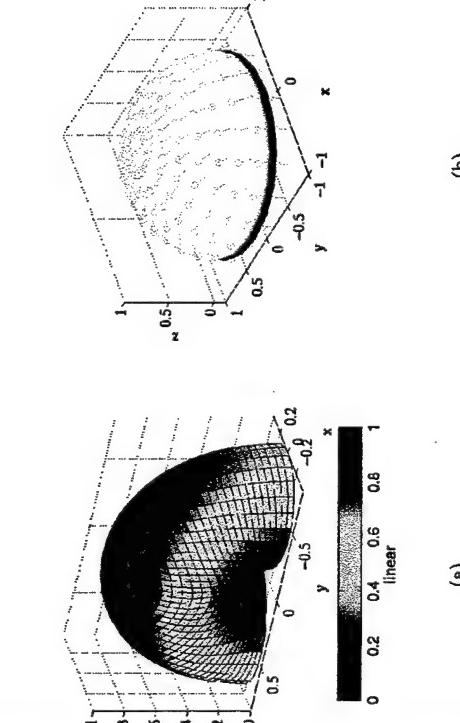
Figure 5.11. The radiation pattern (a) and far-field electric field pattern (b) for a patch antenna. The electric field vectors are plotted tangent to a unit sphere with their tails denoted as circles.

Chapter 4 has a radiation pattern, as shown in Figure 5.11(a) and electric field polarization plot, Figure 5.11(b) for a magnetic current density in the $\widehat{a_x}$ direction.

5.2.3 Numerical Model Structure   *MatLab* 5.3 produced by Math-Works is chosen as the programming tool for a design-oriented lens array modeling tool. A complete listing of the MatLab functions are listed in Appendix A, with a basic description of the objects listed below. Many of the objects and functions have names taken from microwave and optical imaging systems.

Object 1: **antenna**

The antenna object describes the type of antenna, its position and its orientation. Stored with the antenna object is a set of coordinate transformation matrixes that allow easy calculation of coupling between antenna elements per Equation (5.9).

Object 2: **Lens**

The lens object contains all the information to describe the lens: size, shape (circular or rectangular), number of unit cells and the design parameters (F/D and θ_0) as well as a complete description of each unit cell (gain, delay, feed side and non-feed side antenna objects). Since the lens array is an approximation to a lens, the way the aperture is populated with unit cells can effect its performance. To explore the effects of the unit cell placement, two different lattices and aperture approximations are used. The unit cells may be packed with a triangular lattice or maximally fill the aperture. The aperture may be approximated by constraining the unit cells to be within the aperture boundary, or letting them cross the boundary by $\frac{1}{4}$ unit cell width. Figure 5.12 shows four examples of a 3λ radius circular lens with different lattices and aperture approximations.

Object 3: **Array**

For comparison to a lens array, a planar array object is develop under the same conditions as the lens array.

Object 4: **Detector**

A detector is an antenna element (or antenna array with corporate feed) placed behind the used lens to detect the focal image of the lens. Each element in the detector is an antenna object with a delay and gain associated with it. The output of the corporate feed network may also have a gain associated with it.

Object 5: **Imager**

An imager is a collection of detectors used to sample the image produced by the lens.

Object 6: **Source**

There are two different types of sources: near-field and far-field. A near-field source is an antenna object radiating towards the array. A far-field source is a plane wave incident upon the array. Both sources have input magnitudes, I_{in} for near-field and E_{in} for far-field, and phase delays.

Object 7: Channel

The channel describes how the signals for a user is received by the array. It is a collection of near-field and far-field sources that represent the paths that the users signal took to the array.

Object 8: QOLTF

The Quasi-Optical Lens Transfer Function is a matrix that describes the induced open circuit voltages at the outputs of each detector in an imager due to a channel through a lens array.

Object 9: NoiseTF

The Noise Transfer Function is a matrix that relates the noise introduced by gain elements in a lens and noise in the detectors to the outputs of a "imager".

Object 10: Radiation Pattern

Radiation pattern is the far-field radiation pattern of a lens array as seen in reception by a detector feed.

Object 11: Image Pattern

The Image pattern is the received power that a detector with unit load will receive when placed on the focal surface for a given "channel".

For easier interpretation of radiation patterns and image patterns, two different coordinate system are used in their calculation, Figure 5.13. The (α, χ) coordinate system is the negative of the (θ, ϕ) coordinate system.

Combining the above objects makes it possible to model communication systems that have mobile sources, multiple users, multi-path propagation channels, and time varying polarization.

5.3 Lens Simulation Results

In designing planar lens arrays, there are only two degrees of freedom: the relative antenna positions and unit cell delays. These two degrees of freedom

---
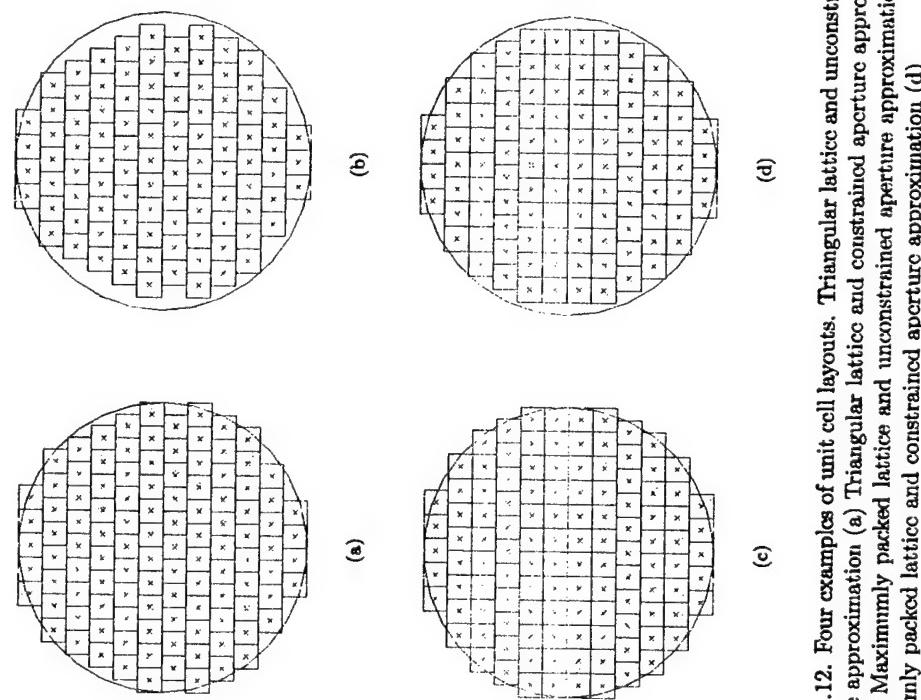
Figure 5.12. Four examples of unit cell layouts. Triangular lattice and unconstrained aperture approximation (a) Triangular lattice and constrained aperture approximation (b) Maximumly packed lattice and unconstrained aperture approximation (c) Maximumly packed lattice and constrained aperture approximation (d)
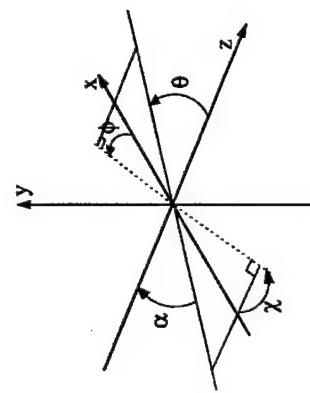
Figure 5.13. The two coordinate systems used to plot radiation and image patterns. By plotting the results in these two coordinate system, the inverse image introduced by the lens is reversed and a source at $(\alpha_s, \chi_s) = (C_1, C_2)$ is detected at $(\theta_d, \phi_d) = (C_1, C_2)$.

are uniquely specified by two design parameters, F/D and cone of best focus, θ_0. These two parameters only define the structure of the lens and do not define the antenna elements nor how the unit cells are placed in the array. To understand how the lens design parameters and unit cell design lens' overall performance, several numerical experiments are performed. The metrics that define a good lens design are application-specific. For example, in a spatial power combining application, the loss in the lens system affects the combining efficiency and is therefore most important. In an angle diversity communication system, the aberrations (distortions in the image) may be more important than the loss. Since it is not known which performance metrics are most important for angle diversity communication systems, a general characterization of focusing ability of the lens is performed for different lens designs. In most communication systems, it is the average performance that matters, and in an imaging system this average is over spatial angles. In the numerical experiments, fifty far-field sources are used, uniformly spread over the field of view for 0° to 90° off axis, Figure 5.14. The sources are arrange on concentric circles of 0° (source 1), 20° (sources 2-7), 40° (sources 8-18), 60° (sources 19-33), and 80° (sources 34-50) off axis.
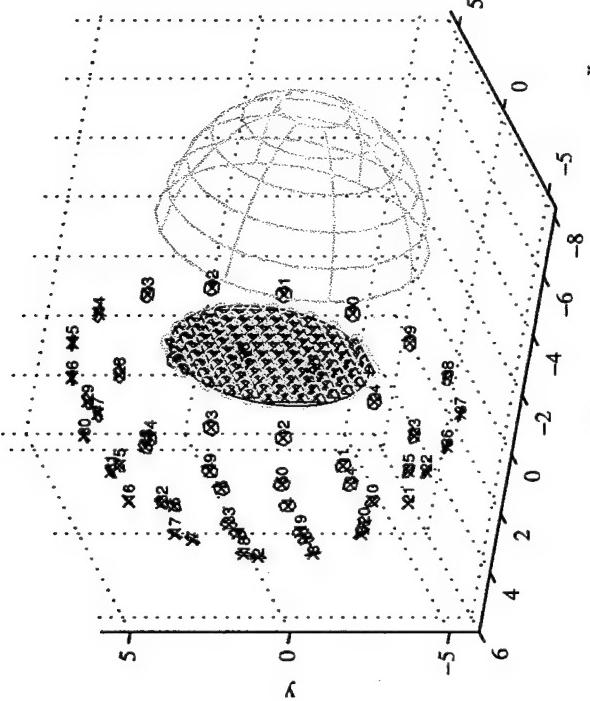
Figure 5.14: Fifty far-field sources used in numerical lens array simulations.

5.3.1 F/D versus θ_0 The first experiment examines the loss in a lens as a function of F/D and θ_0. Eight prototype lens designs are based on all combinations (patch or semi-directional antennas; triangle or maximumly packing lattice; and constrained or unconstrained aperture approximation). The lenses are 4λ in diameter and use $\frac{\lambda}{2}$ by $\frac{\lambda}{2}$ unit cells. For each prototype lens, the F/D and θ_0 are swept from 0.5 to 2 and 0° to 45°, respectively. Since some F/D and θ_0 combinations can lead to lens designs where the feed-sided antennas are outside their respective unit cell boundaries, in each simulation this physical limitation is verified. The loss is calculated for each source as the lens focuses the radiation onto a detector on the focal surface. The loss is the amount of collected power by the detector compared to the total power incident on the array aperture (the area of the lens aperture times

Figure 5.15. The average (a) and variance (b) for the loss of a 4 λ lens with semi-directional antennas over all fifty sources showing the trends in loss as a function of F/D and θ_0.

the incident power flux density). Therefore reduction in antenna element gain for scan angles off the optical axis is considered to be loss.

Since the amount of data obtained in this numerical experiment is large, only the trends will be discussed here. These trends are independent of the antenna elements used, size of the lens and its field of view. However, the optimal design for the lens (the optimal values for F/D and θ_0) are highly dependent on these parameters and the lens design needs to be optimize for each set of parameters. Figure 5.15(a) shows the average loss over the 50 sources as a function of F/D and θ_0 for a maximally packed unconstrained lens with semi-directional antennas. The loss does not monotonically decrease as F/D increases but oscillates. This oscillation is not a result of standing waves between the lens array and detector, since the model does not include these effects. The oscillation most be a result of sinusoidal variation in the phase error as a function of F/D. For a constant F/D, the loss does monotonically decrease as θ_0 increases. The mean loss is minimize by having the smallest possible F/D and θ_0. This is equivalent to having the focal surface as

close as possible to the lens, suggesting that path loss between lens and detector is the dominant loss mechanism. Simulations are performed where path loss and polarization loss are calculated independently. The simulations show that almost all the loss can be account for by the path loss, while the polarization loss contributes only 2% to 5% of the total loss.

The variance of the loss over the 50 sources describes the how much the loss changes as a source moves off axis. Ideally, the loss should remain constant and the variance would be zero. The variance of the loss decreases as the cone of perfect focus moves away from the optical axis, Figure 5.15(b). This is consistent with the phase error for the unit cells being averaged over the field of view (section 5.1).

Assuming the phase error is a polynomial function of scan angle, Shelton [49] suggests that the optimal positions for the perfect focal points in a two-dimensional lens array occur at the roots of a Chebyshev polynomial whose order is the number of degrees of freedom. For example, a McGrath lens has two degrees of freedom, and the points of perfect focus should be placed at $\pm 45°$ for minimum error up to 60° off axis. Figures 5.16(a) and 5.16(b) show the loss for a 4 λ lens with (F/D=.82, $\theta_0 = 14°$) and (F/D=0.5, $\theta_0 = 45°$), respectively. These plots illustrate that it is possible to design a lens with large or small θ_0 and still achieve low variance in loss if the F/D is chosen properly. It appears that the semi-directional lenses have less variation in their loss than patch-antennas lenses. This is probably due to the reduced gain of the patch antenna when the sources are far from the optical axis. It also appears that the patch antenna limits the useful scan angle of the lens to about $\pm 60°$, independent of θ_0. Another interesting phenomenon is the variation in loss for sources that have the same angle from the optical axis (e.g. sources 19-33 that are 60° of axis). The loss increases and decreases in a cosine fashion, showing the affect polarization loss in the system.

The lattice and aperture approximation seems to have little effect. However,

close as possible to the lens, suggesting that path loss between lens and detector is the dominant loss mechanism. Simulations are performed where path loss and polarization loss are calculated independently. The simulations show that almost all the loss can be account for by the path loss, while the polarization loss contributes only 2% to 5% of the total loss.

The variance of the loss over the 50 sources describes the how much the loss changes as a source moves off axis. Ideally, the loss should remain constant and the variance would be zero. The variance of the loss decreases as the cone of perfect focus moves away from the optical axis, Figure 5.15(b). This is consistent with the phase error for the unit cells being averaged over the field of view (section 5.1).

Assuming the phase error is a polynomial function of scan angle, Shelton [49] suggests that the optimal positions for the perfect focal points in a two-dimensional lens array occur at the roots of a Chebyshev polynomial whose order is the number of degrees of freedom. For example, a McGrath lens has two degrees of freedom, and the points of perfect focus should be placed at $\pm45°$ for minimum error up to 60° off axis. Figures 5.16(a) and 5.16(b) show the loss for a $4\,\lambda$ lens with (F/D=.82, $\theta_0 = 14°$) and (F/D=0.5, $\theta_0 = 45°$), respectively. These plots illustrate that it is possible to design a lens with large or small θ_0 and still achieve low variance in loss if the F/D is chosen properly. It appears that the semi-directional lenses have less variation in their loss than patch-antennas lenses. This is probably due to the reduced gain of the patch antennas when the sources are far from the optical axis. It also appears that the patch antenna limits the useful scan angle of the lens to about $\pm60°$, independent of θ_0. Another interesting phenomenon is the variation in loss for sources that have the same angle from the optical axis (e.g. sources 19-33 that are 60° off axis). The loss increases and decreases in a cosine fashion, showing the affect polarization loss in the system.

The lattice and aperture approximation seems to have little effect. However,
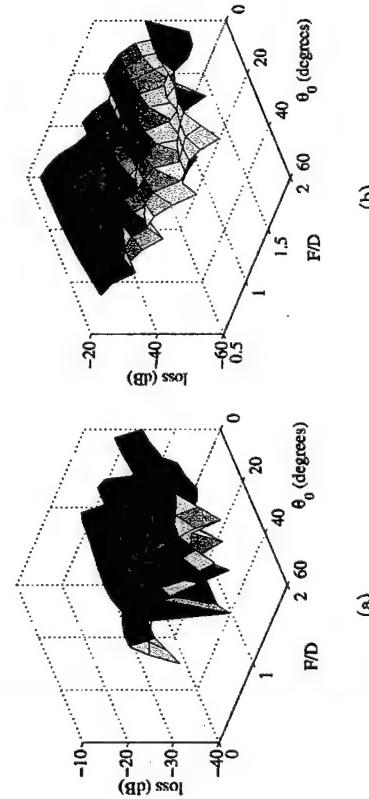
Figure 5.15. The average (a) and variance (b) for the loss of a $4\,\lambda$ lens with semi-directional antennas over all fifty sources showing the trends in loss as a function of F/D and θ_0.

the incident power flux density). Therefore reduction in antenna element gain for scan angles off the optical axis is considered to be loss.

Since the amount of data obtained in this numerical experiment is large, only the trends will be discussed here. These trends are independent of the antenna elements used, size of the lens and its field of view. However, the optimal design for the lens (the optimal values for F/D and θ_0) are highly dependent on these parameters and the lens design needs to be optimize for each set of parameters. Figure 5.15(a) shows the average loss over the 50 sources as a function of F/D and θ_0 for a maximally packed unconstrained lens with semi-directional antennas. The loss does not monotonically decrease as F/D increases but oscillates. This oscillation is not a result of standing waves between the lens array and detector, since the model does not include these effects. The oscillation must be a result of sinusoidal variation in the phase error as a function of F/D. For a constant F/D, the loss does monotonically decrease as θ_0 increases. The mean loss is minimize by having the smallest possible F/D and θ_0. This is equivalent to having the focal surface as

the larger the number of unit cells in the array, the more power it can collect, which argues for maximally packed unconstrained lens designs.

5.3.2 Lens Abberations

The next experiment looks at aberration in the image for different lens designs as the source moves off axis. For an ideal lens, the image produced by a far-field source off axis should be identical to the one produced by a source on axis, only shifted. Aberrations are distortions of the image, as illustrated in Figure 5.17. The mean loss is minimized when F/d and θ_0. The aberrations can reduce the performance of a communication system by spreading the received power over a larger area on the focal surface. This will require more detectors to sample the image, which adds more noise sources to the angle diversity communication system and reduces the directivity of the lens.

Again, eight prototype 4λ lens designs with unit cell size of $\frac{\lambda}{2}$ by $\frac{\lambda}{2}$ are used. Each is a combination of the semi-directional or patch antennas, triangular or maximally packing lattices, and ($F/D=0.8$, $\theta_0 = 14°$) or ($F/D=0.5$, $\theta_0 = 45°$). For each of the prototype lenses, image patterns are calculated for each of the 50 sources. The half power beam radius (HPBR) for each pattern is calculated in eight different directions. The first lobe (ring) is also measured in a similar manner. The metrics of interest are: the average HPBR, the average radius of the first lobe ring and the ratio of the first lobe average power to the main beam power. Another metric is the average difference between the phase at the center of the main beam and the HPBR points. This metric is not important for adaptive systems, but has value for non-adaptive systems and this is discussed in the future work section 7.3.1.

Tables 5.1, 5.2, 5.3, and 5.4 show the metrics for the lens designs. The percent standard deviation (STD) is the percent ratio of the STD to the mean. It gives an estimate of how much the HPBR and first lobe radius distort from the ideal circle. The percent STD for the HPBR phase is the ratio of the STD of delta phase at the HPBR points to 360°. For a uniformly distributed phase from −180° to 180°,
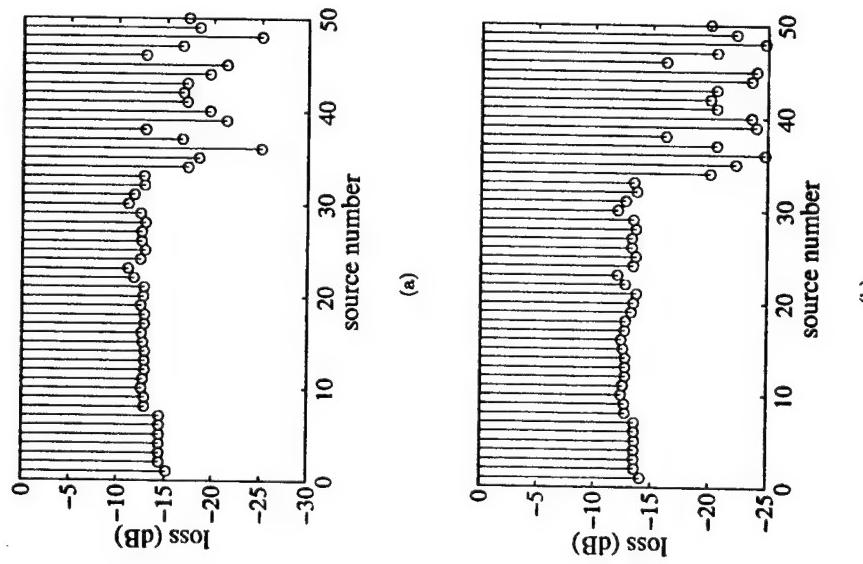
Figure 5.16. The loss for a 4λ diameter lens with semi-directional antennas and design parameters $F/D=0.82$ $\theta_0 = 14°$ (a) and $F/D=0.5$ $\theta_0 = 45°$ (b).

Table 5.1. Metrics for F/D=0.82, $\theta_0 = 14°$, and a triangular lattice using semi-directional antennas.

metric	0°	20°	40°	60°	80°
Average HPBR (degrees)	8.3	8.6	9.50	10.76	9.88
STD (%)	2.2e-14	19.6	13.93	18.76	19.45
Average HPBR Phase	20	18.54	-148.6	1.78	-45
STD (%)	0	14.6	32	22.6	20.92
Average First Lobe Radius (degrees)	27.2	21.14	21.8	21.8	21.85
STD (%)	0	51.5	77.07	86.9	75.23
Average First Lobe Power (dB)	-18	-5.68	-4.10	-3.22	-4.1

Table 5.2. Metrics for F/D=0.5, $\theta_0 = 45°$, and a triangular lattice using semi-directional antennas.

metric	0°	20°	40°	60°	80°
Average HPBR (degrees)	7.9	8.34	9.5	11.84	10.32
STD (%)	0	21.93	24.27	40.34	37.34
Average HPBR Phase	3.5	-16.97	42.56	8.38	15.14
STD (%)	0	2.89	4.59	6.32	5.2
Average First Lobe Radius (degrees)	25.7	23.51	20.85	15.6	19.49
STD (%)	0	62.42	83.24	87.08	79.0
Average First Lobe Power (dB)	-25.77	-5.43	-2.86	-1.77	-3.13

Figure 5.17. Image patterns for sources at 0° (a), 20° (b), 40° (c) and 60° (d) off axis showing the aberrations in the image.

the STD is 104° (or 29%).

Several trends can be deduced from the data. First of all, the comparison between like lenses with triangular lattices versus maximally packed lattices, shows almost identical performances, again arguing for building lenses with as many unit cells as possible. All four lens have the same increase in average HPBR size, starting with about 16° beamwidth at 0° off axis to about 22° beamwidth at 60° off axis. However, the variance on the HPBR is lower for the lenses with $\theta_0 = 14°$, especially for large scan angles. The phase stability for the lenses with $\theta = 45°$ is nearly constant over scan angle, while the lenses with $\theta = 14°$ appear to approach uniform distribution at large scan angles. The average power in the first side lobe approaches the power in the main lobe at large scan angles for all four lens designs, however the lenses with $\theta_0 = 45°$ have better side lobe levels, especially at small scan angles. All four of the lenses have large variations in the first lobe power which is evident from Figure 5.17, where the first lobe power is larger on the side farthest from the center.

5.3.3 Lens Array Loss under Scaling

We have seen in Section 5.3.1 that the loss in lens arrays can be large but it is also important to see if the loss in the lens arrays changes with size. The arguments in section 1.3 for improved SNR in communication systems relies on the assumption that the loss in a lens array does not increase with size. This assumption is based on a geometric argument for a lens operating in the paraxial region. Let us assume that there is an aperture antenna whose effective area equals its physical area and is $2x_0$ by $2y_0$, on the focal surface of a lens with diameter, D, Figure 5.18(a). The angular size of this antenna viewed from the lens is $\theta_{antenna} \approx 2\frac{x_0}{F} = 2\frac{x_0}{C_{const}D}$, where $F/D = C_{const}$. The angle that includes the HPBW spot focused on the lens is $\theta_{HPBW} \approx 2\frac{0.51\lambda}{D}$ [54], Figure 5.18(b). Both the aperture size and the focal spot size scale as the inverse of the lens diameter, resulting in loss independent of lens size.

Two prototype lenses with semi-directional antennas, $\frac{\lambda}{2}$ by $\frac{\lambda}{2}$ unit cells

Table 5.3. Metrics for F/D=0.82, $\theta_0 = 14°$, and a maximally packed lattice using semi-directional antennas.

metric	0°	20°	40°	60°	80°
Average HPBR (degrees)	7.84	8.32	9.2	10.58	9.6
STD (%)	0	20.24	13.2	18.52	19.7
Average HPBR Phase	18.1	17.37	-141.1	2.37	-42.47
STD (%)	0	14.03	32.58	22.32	20.43
Average First Lobe Radius (degrees)	25.7	22.83	21.9	19.04	21.06
STD (%)	0	38.57	75.12	92.25	73.53
Average First Lobe Power (dB)	-17.1	-7.32	-4.14	-2.80	-4.02

Table 5.4. Metrics for F/D=0.5, $\theta_0 = 45°$, and a maximally packed lattice using semi-directional antennas.

metric	0°	20°	40°	60°	80°
Average HPBR (degrees)	7.33	8.12	9.67	11.09	10.56
STD (%)	1.3e-14	23.54	23.4	41.13	37.48
Average HPBR Phase	1.07	-18.7	28.89	1.50	9.9
STD (%)	0	2.54	4	5.8	5.28
Average First Lobe Radius (degrees)	34	23.71	19.5	18.0	18.6
STD (%)	0	69.9	82.0	88.32	86.0
Average First Lobe Power (dB)	-25.53 7	-4.81 7i	-2.71	-2.41	-2.48

(a)

(b)
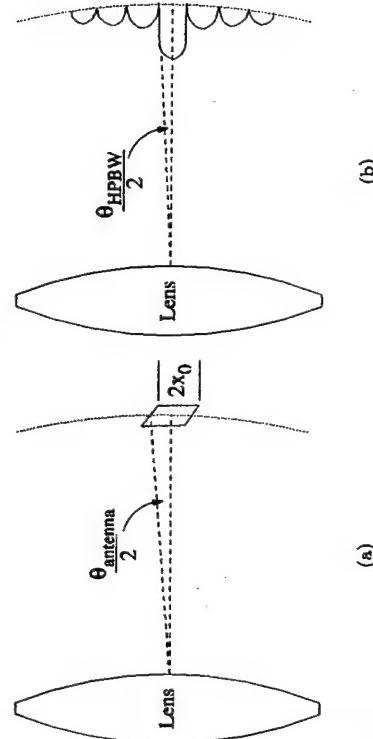
Figure 5.18. Illustration showing the angular size of an antenna on the focal surface (a) and the HPBW spot (b) as viewed from the lens.

spacing and, (F/D=0.8, $\theta_0 = 14^o$) or (F/D=0.5, $\theta_0 = 45^o$) are used to estimate the loss of a lens under scaling. The lenses are scaled from 3λ (35 unit cells) to 10λ (306 unit cells) in diameter. The metric of interest is how the loss scales with the number of unit cells, N_{cells}.

$$loss = N_{cells}^\alpha. \qquad (5.39)$$

By taking $20log_{10}(\cdot)$ of both sides in Equation (5.39), α becomes the slope of the curve.

Figure 5.19 show the average loss for the lens sources. For the lens with F/D=0.8 and $\theta_0 = 14^o$, the loss is constant for sources within 20° of the optical axis.

For $\theta_0 = 14^o$, the phase error is minimized for angles less than 20° from Shelton's approximation, showing the lens array to be a good approximation to a lens. For sources 20° or more off axis, the loss scales as $N_{cells}^{0.11}$ to $N_{cells}^{0.4}$. For the lens with F/D=0.5 and $\theta_0 = 45^o$ the loss is increases for all source angles but at a smaller rate. The loss scales as $N_{cells}^{0.08}$ to $N_{cells}^{0.22}$. In both cases, the average scaling power over all sources within 60° of the optical axis is $N_{cells}^{0.16}$.
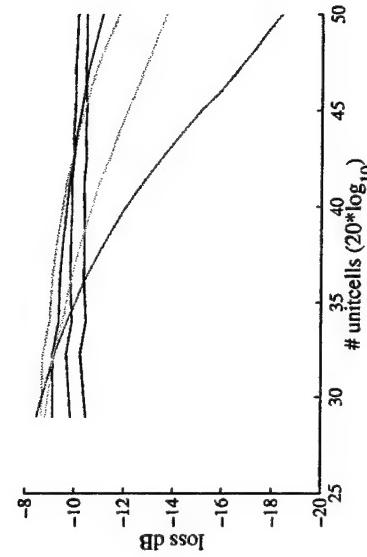
(a)
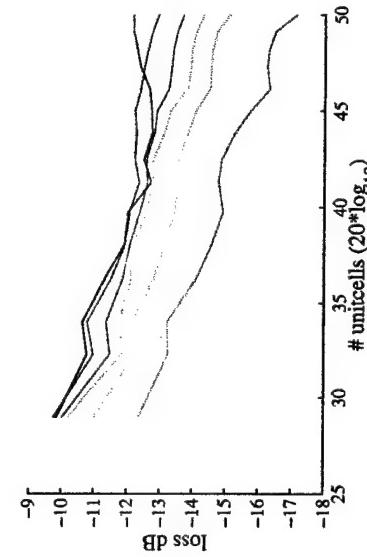
(b)

Figure 5.19. The average loss for a lens with F/D=0.82 and $\theta_0 = 14^o$ (a) and F/D=0.5 and $\theta_0 = 45^o$ (b) lens array as it is scaled from 3λ to 10λ. The source at 0° is in blue, sources at 20° are in red, sources at 40° are in green, sources at 60° are in yellow, sources from 0° to 40° are in magenta and sources from 0° to 60° are in cyan.

CHAPTER 6

SIMULATION OF BEAM-SPACE ADAPTIVE LENS ARRAY SYSTEMS

6.1 The LMS Algorithm

Adaptive arrays communication systems date back to 1950's the when Howells and Applebaum demonstrated the nulling of an interfering signal using a two element array [13]. Shortly afterwards, Widrow developed the LMS (Least Mean Squares) algorithm which is one of the simplest and most common algorithms used in adaptive systems. The LMS algorithm has been used in many different applications like adaptive arrays [18, 15], adaptive noise canceling [55], recursive filters [19], etc. LMS belongs to the class of algorithms that use gradient descent methods to find a minimum on an error surface. Two other popular algorithms, Newton's and steepest descent, also belong to the this class of algorithms.

An excellent tutorial on the LMS algorithm is give in [19], briefly repeated here for convenience. Let us assume that the LMS algorithm is applied to the adaptive array in Figure 6.1. Independent white noise, $n_{i,k}$, is added to each of the antenna signals, $\alpha_i d_k$. The subscript i is used to indicated the i-th signal and the subscript k is used to indicated the k-th sample in time. α_i is a complex coefficient used to describe the amplitude and phase changes of the original transmitted signal d_k that each antenna receives. For convenience, the received signals and noise are expresses in vector form, $d_k A$, and N_k, respectively, where a matrix or a vector are in bold face. The output of the adaptive array, y_k, is a linear weighted sum of the
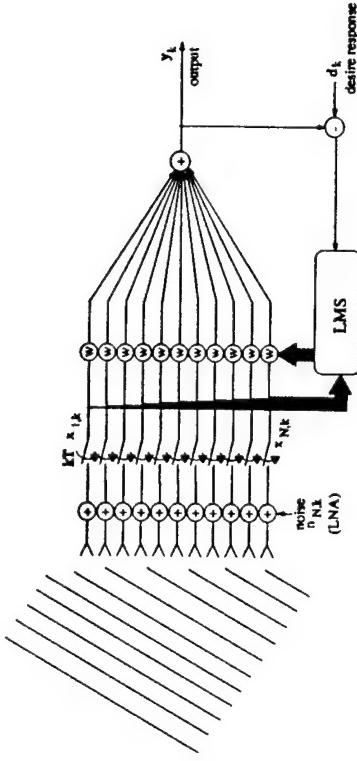
Figure 6.1. An example of a narrowband planar array adaptive receiver. The noise is added after each antenna element to model the effects of the LNAs and switches represent the sampling of the received signals with a sampling period of T.

received antenna signals and noise:

$$y_k = W_k^* X_k, \tag{6.1}$$

$$X_k = d_k A + N_k. \tag{6.2}$$

The vector W_k is a vector contain the weights for the linear sum, while $*$ denotes complex conjugate transpose. W_k, X_k, N_k and A are column vectors.

The LMS algorithm adjusts the weights such that the error,

$$e_k = d_k - y_k, \tag{6.3}$$

is minimized. The mean square error, $E[|e_k^2|]$, is a quadratic function of W:

$$\zeta = E[|e_k^2|] = E[d_k^2] + W^* R W - 2P^*_i W, \tag{6.4}$$

where

$$R = E[X_k X_k^*] \tag{6.5}$$

$$P = E[d_k^* X_k] \tag{6.6}$$

Here the $E[\cdot]$ denotes the expectation. Since the error function, ζ, is quadratic in $\mathbf{W}$, it has one minimum given by

$$\mathbf{W}_{opt} = \mathbf{R}^{-1}\mathbf{P} \tag{6.7}$$

which is sometimes known as the Wiener solution.

The LMS algorithm, just like the Newton Raphson and steepest descent algorithms, use an estimate of the gradient of the error function, ζ, to transverse down the slope to the local minimum. In the LMS case, the algorithm uses the instantaneous value for the squared error as the estimate for ζ. By taking the partial derivative of the $|e|_k^2$ in terms of weights, $\mathbf{W}$, the gradient is estimated to be

$$\widehat{\nabla}_k = \frac{\partial |e|_k^2}{\partial \mathbf{W}} = -2e_k^*\mathbf{X}_k \tag{6.8}$$

The next set of weights is found by using the current set of weights and progressing in the negative gradient direction:

$$\mathbf{W}_{k+1} = \mathbf{W}_k - \mu\widehat{\nabla}$$
$$= \mathbf{W}_k - 2\mu e_k^*\mathbf{X}_k \tag{6.9}$$

The constant μ is a gain constant that regulates the step size to insure convergence.

A convenient way to express the weights is in terms of the optimal weight solution and the weight error variable, $\mathbf{V}_k$,

$$\mathbf{W}_k = \mathbf{V}_k + \mathbf{W}_{opt} \tag{6.10}$$

and examine the expected decay of $\mathbf{V}_k$ by calculating and neglecting noise in the system

$$E[\mathbf{W}_{k+1}] = E[\mathbf{W}_k] + 2\mu E[e_k\mathbf{X}_k]$$
$$= E[\mathbf{W}_k] + 2\mu\left(E[d_k\mathbf{X}_k] - E[\mathbf{X}_k\mathbf{X}_k^*\mathbf{W}_k]\right)$$
$$= E[\mathbf{W}_k] + 2\mu\left(\mathbf{P} - \mathbf{R}\,E[\mathbf{W}_k]\right) \tag{6.11}$$
$$= (\mathbf{I} - 2\mu\mathbf{R})\,E[\mathbf{W}_k] + 2\mu\mathbf{R}\mathbf{W}_{opt}$$

Using the eigenvectors of $\mathbf{R}$ to rotate the system to its principal-axis coordinates (where the signal correlation matrix, $\mathbf{R}$, is diagonal) expressed using a primed variable, it can be shown that

$$\mathbf{V}_k' = (\mathbf{I} - 2\mu\Lambda)^k\mathbf{V}_0' \tag{6.12}$$

where Λ is the diagonal matrix containing the eigenvalues of $\mathbf{R}$ and $\mathbf{V}_0'$ is the initial weight vector. From Equation (6.12), the algorithm learns as a decay of a set of exponential functions. It shows that the mean vector converges slower for more weights. It also shows that for the mean vector to converge, the gain constant μ needs to be

$$0 < \mu < \frac{1}{\lambda_{max}}, \tag{6.13}$$

where λ_{max} is the largest eigenvalue of $\mathbf{R}$. Since the eigenvalues of $\mathbf{R}$ are difficult to calculate, μ is often set to be

$$\mu = \frac{\mu_0}{Tr[\mathbf{R}]} \tag{6.14}$$

where $Tr[\cdot]$ is the trace of $\mathbf{R}$. Since $Tr[\mathbf{R}]$ is the sum of the eigenvalues, this condition guarantees convergence.

Equation (6.12) shows the mean of the weights decay to the optimal solution. The noise in the system causes the weights to dither around the optimal value. The amount that the weights dither depends on the gain constant, μ. Let us assume the gradient estimation $\widehat{\nabla}_k$ has independent noise added, $\mathbf{Z}_k$, such that

$$\widehat{\nabla}_k = \nabla_k + \mathbf{Z}_k \tag{6.15}$$

Assuming that the algorithm has already converged to steady-state solution, $\mathbf{W}_{opt}$, then $\nabla_k = 0$. Then in accordance with Equations (6.15) and (6.8),

$$\mathbf{Z}_k = -2e_k^*\mathbf{X}_k \tag{6.16}$$

The covariance of the noise is given by

$$cov[Z_k] = E[Z_k Z_k^*] = 4E[\|e_k\|^2 X_k X_k^*] \tag{6.17}$$

$$\approx 4\zeta_{min} R$$

When the weights are near the optimal solution, the error e_k is approximately independent of the received signals, X_k.

ζ_{min} is the minimum value for the error function given by

$$\zeta_{min} = E[|d_k|^2] - P^* W_{opt} \tag{6.18}$$

Using the eigenvector matrix of R, denoted Q, to rotate the system to the principal-axis coordinates, we have

$$E[Z_k'] = cov[Z_k' Z_k^{*}] = E[Q^{-1} Z_k Z_k^* Q] \tag{6.19}$$

$$\approx 4\zeta_{min} \Lambda$$

It can be shown that from Equation (6.19), the covariance of V_k, (weight noise), is given by

$$cov[V_k' V_k^{*}] = \frac{\mu}{4}(\Lambda - \mu\Lambda^2)^{-1} E[Z_k' Z_k^{*}] \tag{6.20}$$

$$\approx \mu\zeta_{min}(\Lambda - \mu\Lambda^2)^{-1}\Lambda$$

$$\approx \mu\zeta_{min} I$$

assuming μ is relatively small. From Equation (6.12), each weight relaxes with a time constant

$$\tau_i \approx \frac{1}{2\mu\lambda_i} \tag{6.21}$$

showing the natural tradeoff between the weight noise and adaptation rate.

6.2 Sampling the Lens Array Focal Surface

To insure that all the information contained in the image is preserved, the detectors are placed on the focal surface in accordance with the two-dimensional sampling theorem. The sampling theorem dictates the sampling period and pattern for a finite bandwidth function that is linear and shift invariant. Given a function $f(x,y)$ and its two-dimensional Fourier transform $F(\omega_x, \omega_y)$, then the sampled version of $f(x,y)$, $f(n,m)$, given by

$$f(n,m) = f(x,y)|_{(x,y)=V(m,n)} \tag{6.22}$$

$$\tag{6.23}$$

where (m,n) samples are given by,

$$\begin{bmatrix} x \\ y \end{bmatrix} = [v]\begin{bmatrix} n \\ m \end{bmatrix}$$

$$= [V_1 V_2]\begin{bmatrix} n \\ m \end{bmatrix} \tag{6.24}$$

where V is a two dimensional matrix containing the vectors V_1, V_2 for the sampling pattern. An example of sampling is given in Figure 6.2 when $f(x,y)$ is a circle function, which is a hexagonal pattern. The Fourier transform of $f(n,m)$ is the sum of the aliased copies of $F(\omega_x, \omega_y)$ given by

$$\mathcal{F}\{f(n,m)\} = \frac{1}{|det V|}\sum_{k,l} F(\omega_x - \omega_{xo}, \omega_y - \omega_{yo}) \tag{6.25}$$

$$\begin{bmatrix} f_{x_o} \\ f_{y_o} \end{bmatrix} = [U]\begin{bmatrix} k \\ l \end{bmatrix}, \tag{6.26}$$

$$\tag{6.27}$$

where V is related to U as

$$U^T V = 2\pi I \tag{6.28}$$

From [54], a lens in its paraxial region approximates the two dimensional Fourier transform. If a plane wave is assumed to be the source, then the lens size and

(a)

(b)
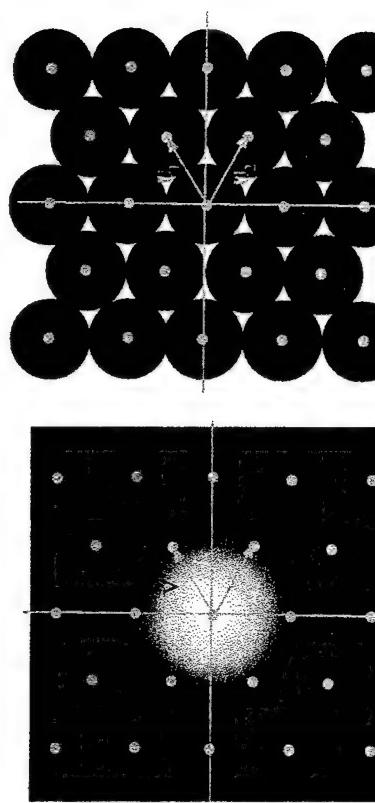
Figure 6.2. An example of two-dimensional hexagon sampling of an Airy pattern (a) and the aliases Fourier transform (b).

shape (pupil function) determine the image pattern. The lens shape is the inverse Fourier transform of the image pattern. The sampling vectors of the image must be short enough to prevent the aliased copies of the pupil function from overlapping.

For a rectangularly shaped lens, Figure 6.3, of size $2A_x$ by $2A_y$, the image intensity pattern is

$$I_0 \propto \text{sinc}^2\left(\frac{2A_x x}{\lambda z}\right) \text{sinc}^2\left(\frac{2A_y y}{\lambda z}\right) \quad (6.29)$$

The sample vectors for the rectangular lens are

$$\mathbf{V} = \begin{bmatrix} \dfrac{\pi\lambda z}{2A_x} & 0 \\ 0 & \dfrac{\pi\lambda z}{2A_y} \end{bmatrix} \quad (6.30)$$

Combining Equations (6.29) and (6.30) gives the samples for the image function

$$I_0(m,n) \propto \text{sinc}^2\left(\frac{2A_x m\pi\lambda z}{2\lambda z A_x}\right) \text{sinc}^2\left(\frac{2A_y n\pi\lambda z}{2\lambda z A_x}\right) \quad (6.31)$$

$$\propto \text{sinc}^2(\pi m)\,\text{sinc}^2(\pi n)$$

For a source on the optical axis, the samples fall on the center of the main lobe and in its nulls, as shown in Figure 6.4.

(a)

(b)
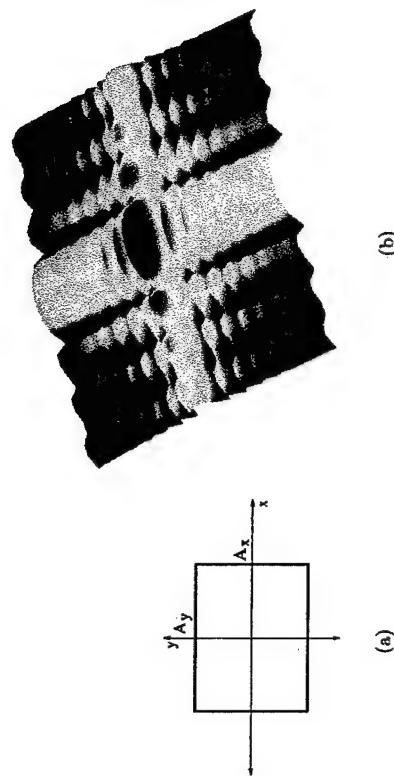
Figure 6.3. A rectangular aperture (a) and its crossed $\text{sinc}(x)$ Fourier transform (b).
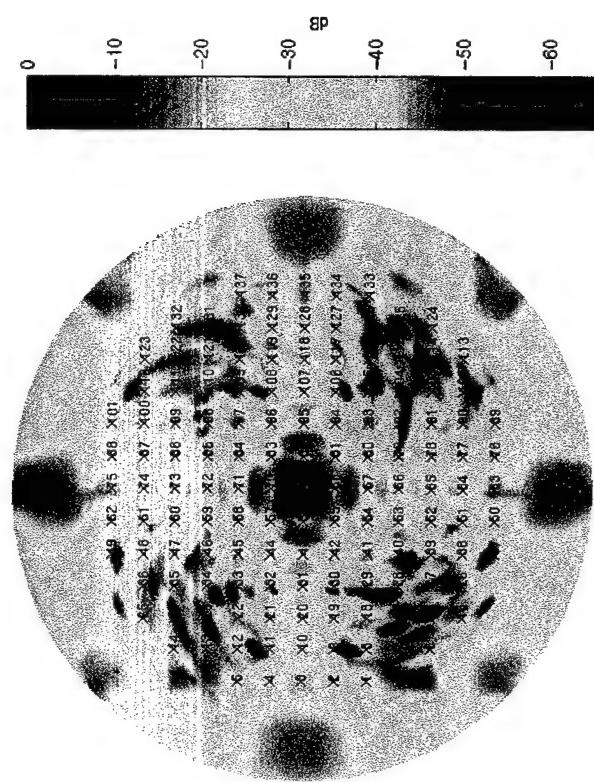
Figure 6.4. The sampling (numbered x's) of a rectangular aperture lens intensity image for a 60° field of view.

(a)

(b)

Figure 6.5: A circular aperture (a) and its Airy pattern Fourier transform (b).

For a circular lens with radius r the image pattern is

$$I_0(\rho) \propto \frac{2J_1(2\pi r\rho)}{2\pi r\rho} \quad \rho = \sqrt{x^2+y^2} \qquad (6.32)$$

and J_1 is the Bessel function of the first kind, Figure 6.5. Equation (6.32) is referred to as the Airy pattern. A hexagonal pattern is the optimal sampling pattern that minimizes the number of samples [56] for a circular aperture with sampling vectors

$$\mathbf{V} = \frac{1}{2r}\begin{bmatrix} 1 & \frac{1}{\sqrt{3}} \\ 1 & -\frac{1}{\sqrt{3}} \end{bmatrix} \qquad (6.33)$$

For a source on the optical axis, the samples also fall on the center of the main lobe and near the nulls, Figure 6.6. It is interesting to note that for both the rectangular and circular lenses, the ratio of the HPBW and the sampling vector lengths is 1.13.

Simulations of the lens array image patterns agree well with Equations (6.29) and (6.32) in the paraxial region, however, it is important to point out that not all of the conditions for the sampling theorem are met by the lens array system: the lens system is noisy, and the aberration in the lens cause the system to be shift-variant. The noise has theoretically infinite bandwidth which contradicts the finite bandwidth condition. However, the noise in the simulations is added after
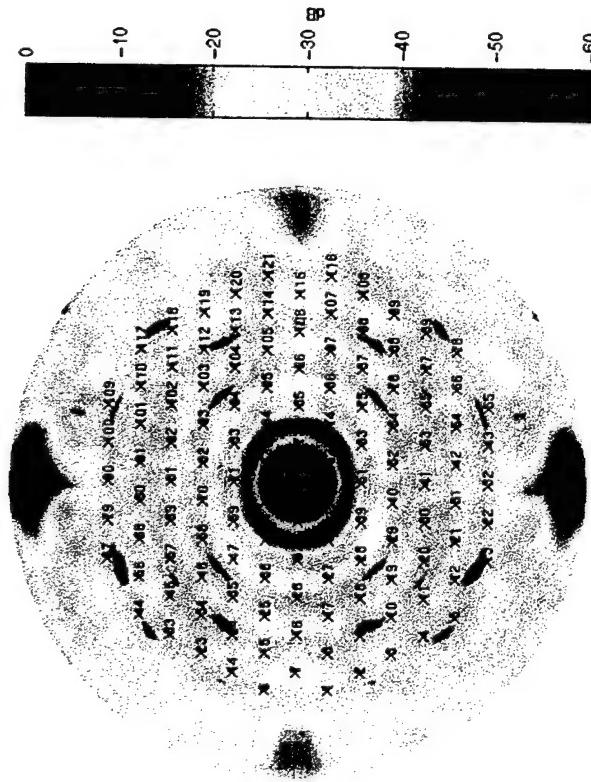
Figure 6.6. The sampling (numbered x's) of a circular aperture lens intensity image for a 60° field of view.

the image for the desired signal is formed, and it is the information in the image pertaining to the desired signal that needs to be retained. The beamwidth of the image increases as the source moves off axis, suggesting that the effective bandwidth of the system decreases. Therefore the detectors oversample the image for signals far off the optical axis. The detectors are placed in an hexagonal pattern on the projection of the focal surface onto a disk, Figure 6.6. When the position of the detectors is inverse transformed back to the focal surface, the outer detectors end up closer together than those in the center, resulting in over sampling of the sources that are far off axis. This projection results in the detectors being placed closer together on the edges of the disk than in the center of the disk, Figure 6.7, again causing over sampling of signals far of the optical axis.
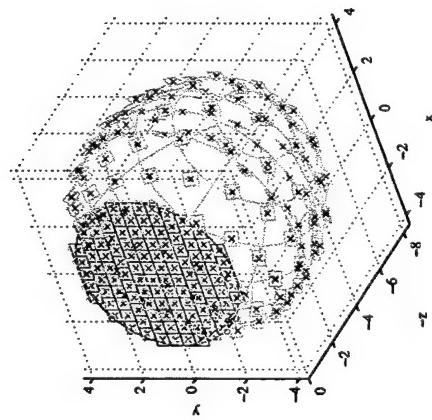
Figure 6.7: An example of a 6λ diameter lens with a sampled focal surface

6.3 Numerical Experiments on Lens Array Systems

The goal of using a lens array as the front end for an adaptive array system is to show that a lens array system has better SNR under scaling and adapts faster since it requires fewer weights. A set of numerical experiments is performed to compare a lens array to a planar array in an adaptive system. For the lens array system, the signals for the detectors are ranked from strongest SNR to weakest SNR. Then the lens system is simulated using the strongest signal, followed by the two strongest signals, etc. For simplicity, the LMS training signal d_k is set equal to the transmitted data.

The output SNR is the principle metric for a communication system. Three different techniques are used to estimate the output SNR of the system. The first is a measure of the bit error rate (BER). This is the average number of wrong guesses a system makes in decoding the information . These experiments use QPSK complex baseband modulated data for the transmitted signals and the algorithm samples at twice the symbol rate. For QPSK modulation, the decisions boundaries are shown in Figure 6.8. The second technique is to estimate the noise in the output signal as the
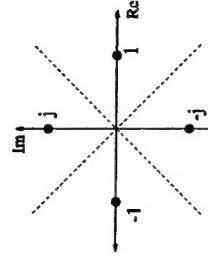
Figure 6.8: QPSK modulation and slicer decision regions.

part of the output signal that is not the transmitted data, d_k. From Equation (6.3), the algorithm tries to make the output signal look like the training signal (data). Therefore the noise at the output is

$$y_{noise,k} = y_k - d_k \quad (6.34)$$

Then the SNR is given by

$$SNR_{data} = \frac{var[d_k] + E[d_k]^2}{var[y_{noise,k}] + E[y_{noise,k}]^2} \quad (6.35)$$

For unit power QPSK modulation, the $var[d_k] = 1$, $E[d_k] = 0$ and equation 6.35 reduces to

$$SNR_{data} = \frac{1}{var[y_{noise,k}] + E[y_{noise,k}]^2} \quad (6.36)$$

The third technique is based on representing the output as the product of the received signals and the weights:

$$y_k = W_k^* X_k \quad (6.37)$$

$$= (W_{opt} + V_k)^* (d_k A + N_k) \quad (6.38)$$

$$= W_{opt}^* d_k A + V_k^* d_k A + W_{opt}^* N_k + V_k^* N_k, \quad (6.39)$$

where the first term is the signal and the last three terms are the noise. This SNR is calculated in two different ways. One uses the calculated W_{opt} given by Equation (6.7) and the second uses the estimate of the optimal weights, $\widehat{W_{opt}} = E[W_k]$, as

Figure 6.9. Sampling of a 6λ diameter circular lens with a source 20° off optical axis.

cells. Semi-directional antennas are used, $F/D = 0.5$, and $\theta_0 = 45°$ to minimize aberrations. The source of the signal is 20° off axis with detectors placed as shown in Figure 6.9. Figure 6.10 shows the distribution of normalized received signal power across all the detectors. The only noise sources in the system are at the detectors. Simulations are run such that the SNR of the peak received signal at the detector, SNR_x, varies from 0.1dB to 10dB, by adjusting the power of the noise sources. The gain constant for the system is determine by Equation (6.14), with $\mu_0 = 0.05$.

Figure 6.11 to 6.13 show the simulated BER, SNR_{data} and SNR_w for the different cases of SNR_x. Under all three metrics, the system performance improves as the SNR_x increases. This seems like an obvious result, but it gives important insight to the operation of the Wiener solution. In the Wiener solution, the weights

---

in Equation (6.12). Then the noise is calculated as

$$y_{noise,k} = y_k - W_{opt}^* d_k A \qquad (6.40)$$

$$\widetilde{y_{noise,k}} = y_k - \widetilde{W_{opt}^*} d_k A \qquad (6.41)$$

and the SNR is

$$SNR_{W_{opt}} = \frac{cov[W_{opt}^* d_k A] + E[W_{opt}^* d_k A]^2}{cov[y_{noise,k}] + E[y_{noise,k}]^2}$$
$$= \frac{|W_{opt}^* A|^2}{cov[y_{noise,k}] + E[y_{noise,k}]^2} \qquad (6.42)$$

$$SNR_{\widetilde{W_{opt}}} = \frac{cov[\widetilde{W_{opt}^*} d_k A] + E[\widetilde{W_{opt}^*} d_k A]^2}{cov[\widetilde{y_{noise,k}}] + E[\widetilde{y_{noise,k}}]^2}$$
$$= \frac{|\widetilde{W_{opt}^*} A|^2}{cov[\widetilde{y_{noise,k}}] + E[\widetilde{y_{noise,k}}]^2} \qquad (6.43)$$

for QPSK modulation. As a comparison for the algorithms performance, the theoretical output $y_{k,opt}$ is calculated using the optimal weights, W_{opt}, as in Equation (6.1).

6.3.1 SNR vs. Number of Signals   The first experiment focuses on the change in SNR as more received signals are added to the lens array system, or in other words, more detector signals are used. Given the arguments in section 1.3, the SNR should increase as more signals are added, until the SNR of the next signal is so small that it reduces the system performance. From Equation (6.39), as each signal is added to the system, the vectors W_{opt}, A, V_k and N_k increase by one. Since the signals are added in decreasing order of SNR, at some point the added value to A is very small. Since the value added to W_{opt} is proportional to the value added to A, Equation (6.7), the new value in W_{opt} is also very small. Thus the first three terms in Equation (6.39) do not change significantly for the added signal. However, the fourth term is independent of A and should add a significant amount of noise to the system.

The simulations are performed using a 6λ diameter lens that has 110 unit

Figure 6.12. The SNR_{data}, Equation (6.35), for SNR_x=0.1, 0.178, 0.316, 1, 1.78, and 3.162 (top to bottom). The adaptive lens array systems is shown in blue and the optimal weight solution is shown in red.

Figure 6.13. The SNR_w for SNR_x= 0.1, 0.178, 0.316, 1 1.78, and 3.162 (top to bottom). The $SNR_{W_{opt}}$ (6.42) for lens array systems is shown in blue, the $SNR_{\widehat{W_{opt}}}$ (6.43) is shown in green and the optimal weight solution is shown in red.

Figure 6.10. The distribution of received signal power across the 121 detectors for the image in Figure 6.9.
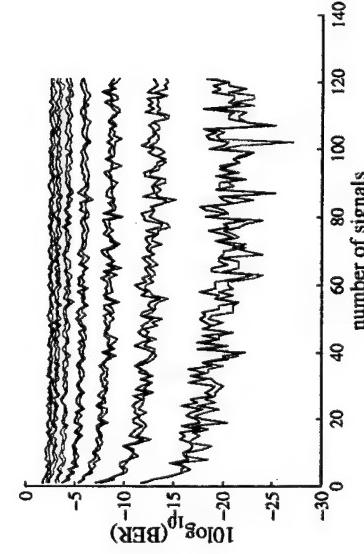
Figure 6.11. The BER for SNR_x 0.1, 0.178, 0.316, 1, 1.78, and 3.162 (top to bottom). The BER for the adaptive lens array systems is shown in blue and BER for the optimal weight solution is shown in red. The noise in the data is due to different noise and data realizations for the different simulations.

noise to the output, for step size, $\mu = \frac{0.05}{\text{Tr}[R]}$.

In all cases, the system performance improves as more signals are added to the system, which is expected. The added signal and noise powers cause the weights' magnitude to adjust inwards. Figure 6.15 shows the change in the optimal weights for the $SNR_x = 0.1$ and $SNR_x = 10$ as more signals are added to the system. In the $SNR_x = 10$ case, the signal power dominates the weight scaling when a few signals are used, but the noise quickly begins to dominate. This is why the high SNR_x case weights scale faster than the low SNR_x case where the noise always dominates. The weights are constantly being changed as the additional signals are added so that the output SNR is a monotonically increasing function. Another way of looking at the weights are diagrams for the output signals. For a system with good SNR, the output signal should form clusters on the modulation constellation diagrams, Figure 6.16. As the noise increases, the center of the clusters move inwards. Figure 6.17 shows the centers of the clusters and there STD radii for various SNR_x. It is interesting to point out that the STD radius do not vary significantly for different numbers of signals used.

Figure 6.18 shows the weight noise power for the different SNR_x cases and variance on the weight noise power which agrees well with calculations based on Equations (6.20), Figure 6.19. The simulated noise power is about 7 dB lower than that calculated and may be related to the fact that the samples of the modulated data are not a uniformly distributed, but are cyclostationary [57, 58]. It is interesting to point out that the weight noise increases for larger SNR_x. Since the increased SNR_x means a decrease in total noise power and input power for the system, Equation (6.5), the gain constant is larger, Equation (6.14), accounting for the increase in weight noise. The variation in weight noise between the weights is small and consistent with theory. These simulations still raise the question: "Is it possible to have a signal whose SNR is so small that the addition of this signal to

Figure 6.14. The Wiener solution for the weights with received signal in Figure 6.9 and SNR_x=0.1, 0.178, 0.316, 0.562, 1, 1.78, 3.162, 5.62, and 10. Each weight track starts with the lowest SNR_x and ends with the largest SNR_x at the weight number.

arc are scaled and decorrelated (R^{-1}) version of the received desired signal (P). The scaling of the weights depends on the relative magnitudes of the noise power to the signal power, and the received signals is found as

$$R = E[X_k X_k^*] \tag{6.44}$$
$$= E[(d_k A + N_k)(d_k A + N_k)^*] \tag{6.45}$$
$$= \sigma_{signal}^2 A A^* + \sigma_{noise} I \tag{6.46}$$
$$= Q^{-1}(\sigma_{signal}^2 A A + \sigma_{noise}^2 I) Q \tag{6.47}$$

For a strong input SNR_x, the weights illuminated by the desired signal have the largest magnitude. However, as the noise in the system increases, the weights adjust inwards to minimize the amount of noise at the output, Equation (6.47) as illustrated by Figure 6.14. The measured performance of the adaptive system is slightly less that that of the optimal weight system showing the effects of the weight noise in the system. Since the difference between the optimal weight system SNR and the adaptive weight system SNR is small, the weight noise does not contribute considerable

Figure 6.16: Eye diagram for adaptive lens array system with $SNR_X = 10$.
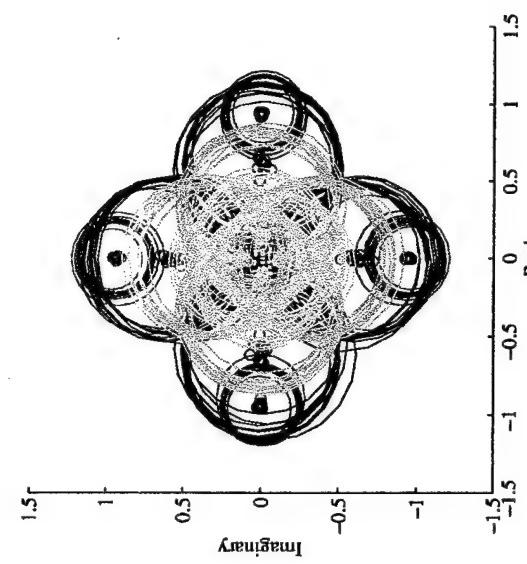
Figure 6.17. Eye diagram for adaptive lens array system with $SNR_X = 10$ (red), $SNR_X = 1$ (blue), $SNR_X = 0.3$ (green) and $SNR_X = 0.1$ (yellow). The centers (o's) and STD rings are plotted for 1, 2, 3,..., 10 and all (121) signals used.

(a)

(b)

Figure 6.15. The Wiener solution for the weights with received signal in Figure 6.9 with $SNR_X = 0.1$ (a) and $SNR_X = 10$ (b) as signals are added to the adaptive lens system. Each point in the weight tracks represents the Wiener solution for 1, 2, 3,..., 10, all (121) received signals used, starting with 1 signals and ending with all signals (the numbered point).

154

Figure 6.19. The calculated weight noise (a) and variation in weight noise between the weights (b) for $SNR_X = 0.1$ (magenta), 0.17 (cyan), 0.32 (red), 0.56 (green), 1 (blue), 1.78 (yellow), 3.162 (magenta), 5.6 (cyan) and 10 (red).

153

Figure 6.18. The measured weight noise (a) and variation in weight noise between the weights (b) for $SNR_X = 0.1$ (magenta), 0.17 (cyan), 0.32 (red), 0.56 (green), 1 (blue), 1.78 (yellow), 3.162 (magenta), 5.6 (cyan) and 10 (red).
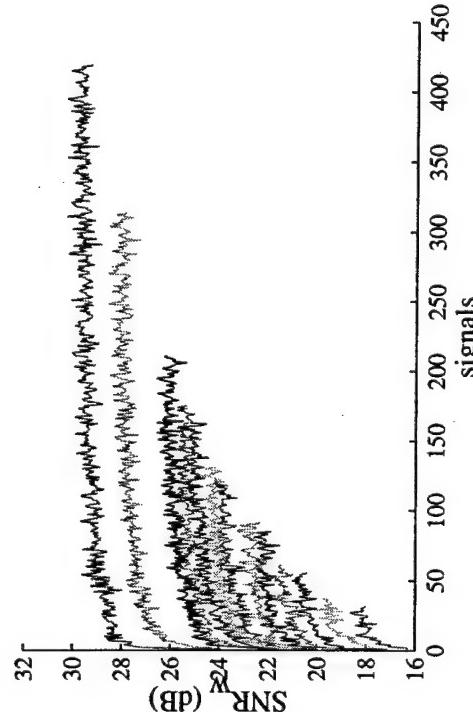
Figure 6.20. The $SNR_{\widehat{W_{opt}}}$ for increasing lens array diameters (3λ, 3.5λ, 4λ, 4.5λ, 5λ, 5.5λ, 6λ, 6.5λ, 7λ, 7.5λ, 8λ, 10λ and 12λ, bottom to top).

the system would reduce the overall performance?". In these simulations, the signals have a range of 30 dB in signal power and the output SNR appears to be a monotonically increasing function. Since the step size changes in accordance to the amount of signal and noise power, the algorithm is still able to estimate the signal components in all of the detectors. We can see the improved estimation of the signals by the reduction in weight noise, Figure 6.18.

6.3.2 SNR vs. Lens Array Size The next experiment focuses on the change in output SNR as the lens size increases. The same basic setup is used in the simulations as in section 6.3.1 with a lens of $F/D = 0.5$, $\theta_0 = 45°$ and a diameter varying from 3λ (28 unit cells) to 12λ (454 unit cells). The detector SNR is set at $SNR_x = 112$. Figure 6.20 shows each system $SNR_{\widehat{W}}$ as the number of signals increases. These curves have the same monotonically increasing nature as signals increases. Figure 6.21 shows the maximum SNR for the lens array and those in section 6.3.1. It is expected that the lens array SNR would grow faster than the
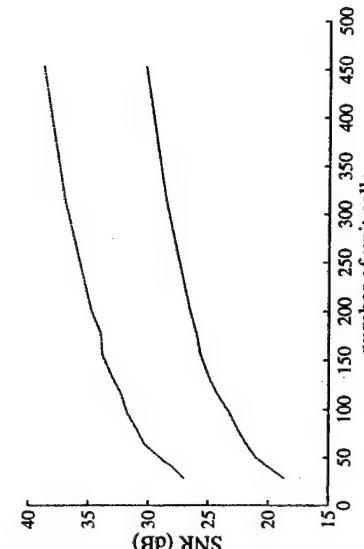
Figure 6.21. Maximum $SNR_{\widehat{W_{opt}}}$ for adaptive lens array (blue) and planar array (red).

SNR for planar array as the lens size increases. Instead, the difference between the two curves appears constant, which suggests the loss increasing with size does have a large effect on performance, Section 5.3.3.

6.3.3 Adaptation Rate Vs. Number of Signals The final asserted advantage of the lens arrays in adaptive systems is its increased adaptation rate due to fewer weights. The same setup is used as in section 6.3.1 to estimate the adaptation rates for a lens array with $SNR_x = 1, 3$ and 10. For each configuration of the system, 500 simulations are performed averaging their error sequences. Figure 6.22 shows the error curves for the $SNR_x = 10$ case with the planar array adapting almost as fast as the lens cases with 9 signals. The lens case where all signals are used is slower than the planar array indicating the weights corresponding to low received signal power need a long time to adapt, as in Equation (6.21). Table 6.1 show the estimated adaptation rates for different lens and planar array systems. Since the step size changes as the number of signals are used, the resulting SNR always increases at the cost of adaption rate. This is a classic tradeoff in adaptive systems and needs to be explored further.

CHAPTER 7

FUTURE WORK AND CONCLUSIONS

7.1 Optically Controlled Lens Arrays

7.1.1 Future Work for Optically Controlled Lens Arrays
There are three principle problems that have to be overcome for optically controlled lens arrays to become practical. First, the size of the SPDT switches should be reduced. Second, a better optical circuit for delivering optical signals to the photodiodes needs to be designed. Third, the overall construction complexity should be minimized further..

Looking at the layout of the unit cell, most of the SPDT switch size comes from the packages of the MESFETs and photodiodes. Using chips or MMICs, the switch size could be reduced by about a factor of 2. If a MMIC SPDT switch is optically controlled through its bias, the size would be reduced further. An all-MMIC solution for the optical controlled microwave switch would significantly reduce the size and complexity of the array manufacturing, but require a special process..

A significant amount of the effort and complexity in the array is in the design and fabrication of the optical fiber mount. By using ridge waveguides [59] stamped or etched into a translucent material to replace the optical mount and the fibers, the optical system can be made in a single fabrication step. In Chapter 4, the electrical and optical circuit could be separated at the photodiode and optical fiber interface. It is also this interface that causes most of the lost in the optical circuit. Instead of separating the two circuits at the photodiode/fiber interface, it would be

Figure 6.22. Error curve tracks for 5 (blue), 9 (red) and all (yellow) signal used in adaptive lens array compared to a planar array (green).

Table 6.1. The number of time steps required for planar and lens array systems to adapt.

SNR_x	Number of Signals in Lens Array				array
	1	3	5	9	all	
1	10	15	30	40	120	70
3.2	8	13	18	27	80	60
10	13	20	22	25	80	40
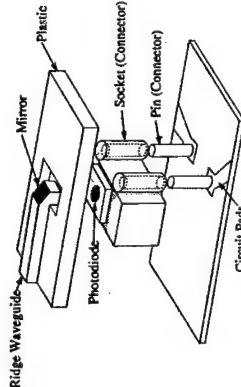
Figure 7.1: New optical feed system for optically controlled arrays.

more efficient to have the connection at the photodiode/MESFET interface, as illustrated in Figure 7.1. By directly gluing the photodiodes to the optical waveguides, the loss in the optical circuit could be reduced to about 1 dB. The photodiode is connected to the MESFETs through a connector, shown as a pin/socket connector in the figure. Since most of the spectral energy in the switch currents is 500 MHz or below, the added inductance and capacitance of a pin/socket connector should have negligible affect.

7.1.2 Conclusion for Optically Controlled Lens Arrays

In this thesis, a high-speed optically controlled T/R active antenna is demonstrated. This active antenna is designed as an element of a $6\lambda_0$ by $3\lambda_0$ cylindrical active lens array with a F/D=1, a directivity of 20 dB and a 10-degree beamwidth in the focusing plane. The lens array is capable of phase-shifterless beamforming in both transmission and reception.

The optically controlled SPDT switch used to route the signal in the active antenna exhibits 0.31 dB insertion loss, 36 dB isolation and -10 dB return loss from 8.36 to 10.8 GHz. Unlike in previously-demonstrated optically-controlled microwave switches, these microwave parameters do not change with the level of incident control (optical) power. However, the optical power level conveniently controls the switching speed alone, making the same switch design easily integrated into applications with different switching speed requirements. Microsecond switching required in most T/R

applications can be accomplished with only microWatts of optical power, but some applications, such as phase shifters in phased array, polarization switching in multi-path environments and wide bandwidth systems would benefit from nanosecond switching speeds. Even though the 2.6 ns speed demonstrated in this paper requires relatively large optical power (about 10 mW), the optical energy is quite low, 21 pJ. For comparison, the fastest reported MEMS microwave switch has a rise time of about 1 μs to 5 μs and requires about 2 nJ of control energy [29]. The switch presented here draws more electrical power (3 mA per PIN diode) than a MEMS switch, but the required DC current is a small fraction of the 160 mA drawn by the LNA and PA. However, the optically controlled microwave switch requires significantly less control energy than a MEMS switch, since the energy distribution is fundamentally different. In some respects, it is easier to generate fast optical pulses than to generate fast high voltage (30 V - 50 V) electrical pulses required for MEMMS switches.

The active array element is designed with off-the-shelf components not optimized for speed or low power, and the coupling of light from the fibers is not optimal. Therefore, the presented results are by no means fundamentally limited and we expect that significant improvements can be made by using PDs with better placement tolerances, using chip MESFETs for the pin diode bias control and by improving the laser diode to photodiode coupling efficiency (using, e.g. microlenses and printed optical waveguide structures). Ultimately, a large portion of the switch, and in principle the entire switch circuit, can be implemented monolithically.

7.2 Modeling of Lens Arrays

7.2.1 Future Work in Modeling Lens Arrays

Most of the errors in the numerical model in Chapter 5 are due to only accounting for direct radiation. Due to the close proximately of the lens array elements and the detectors to the lens, mutual coupling will play an important role in the lens array performance.

pattern presented in section 6.2 appear to corresponds with the beams produce by network lenses. Finally, an analytical model for the lens array transformation should be developed that describes the lens array loss, aberrations and antenna element effects so that a communication model can be developed.

7.2.2 Conclusion for Modeling of Lens Arrays

The preliminary work done in chapter 5 shows that design of lens arrays can be optimized for different performance criteria (i.e. loss, aberration, etc.). Depending on the choice of design parameters, the loss or aberrations in the lens can be optimized. It is still unknown what characteristic is most important to lens array communication systems and more work needs to be done to answer this question.

7.3 Adaptive Lens Array Systems

7.3.1 Future Work for Adaptive Lens Array Systems

The work presented in chapter 6 is for the simplest communication problem, a single user. More work needs to be done in determining the affects of loss and aberrations on the pre-noise beam-forming process gain. Also the tradeoffs between adaption rate and weights noise needs to be explored and compared to a planar array system. The simulations should be expanded to more complicated environments like the multi-user and multi-path environments. Besides simulating more complicated environments, the receiver model should be modified to incorporate practical limitations, such as imperfect training signals. Lens array systems may also have advantages with other algorithms such as angle of arrival algorithms.

7.3.2 An Arguement for Improved SNR with Lens Arrays

Lens arrays may have an advantage over planar arrays by performing some beam-forming before noise in added by gain devices such as low noise amplifiers (LNAs) and down

Figure 7.2: Mutual coupling for patch antenna with arbitrary orientations.

In physical measurements with real active lens arrays, standing waves have been noticed between the lens and the detector elements that can not be explained by simple radiation, [1]. In an attempt to estimate some of the mutual coupling, the numerical model is modified to include the calculation of induced currents in each antenna due to other antennas in its neighborhood, [60, 61, 62]. [62] shows that the mutual coupling between arbitrary orientated patch antennas, Figure 7.2, can be estimated from several full-wave simulations and the following equation

$$|S_{21}| = \frac{\rho}{x^{1+\sin\theta}},$$ (7.1)

where ρ is a fitting parameter. Currents induced by mutual coupling can be calculated from the mutual coupling impedances. With these current, the mutual coupling image is calculated to estimate the error in the image calculations. Averaging the error over all fifty sources (section 6.3) the mean error is about 1.5% suggesting that the image pattern is at best accurate to 18 dB below the main lobe. Hopefully, in the near future it will be possible to perform a full wave model of a lens array.

Besides improving the accuracy of the model, work needs to be done to determine the best lens array design for both circular and rectangular lenses. A question of whether of not the lens array transformation fundamentally limits the achievable radiation patterns for a lens array should be addressed. Work in [25, 26, 24] on the limitations of network lenses may be applicable to lens arrays since the sampling

Figure 7.3. An example of a narrowband planar array adaptive receiver shown for the single user case. The received signals, $s_i(t)$, are amplified with LNAs and down-converted (mixer, local oscillator and LPF) before being sampled (represented by switches). For this model, the effect of the LNAs and down-converters is the addition of noise. The received signals plus noise are sampled with a sampling period T, giving signals, $x_{i,k}$, used by the algorithm. The algorithm uses the training signal, d_k, to adjust the weights, W, such that the SNR at the output, y_k, is maximized.

converters which translate the modulated signal to a lower frequency. In most "standard" adaptive systems, there is a LNA place directly after each array element, Figure 7.3. By placing the amplifiers close to the array elements, the affect of the noise introduced by cable loss can be minimized. In systems that are not limited by "sky" noise, the noise generated by the LNAs becomes the dominate noise source and is usually considered to be white and uncorrelated. After the LNA, a down-converter are usually added to mix the carrier signal to a lower frequency which is easier to process. Down-converters, which contain mixers (multipliers), local oscillators (LOs) and low pass filters (LPFs) do add additional noise and can have gain greater than

unity. For our purpose a complex-baseband model will be used and the noise generated by a LNAs and down-converters are modeled as uncorrelated and white noise sources. The output signal, y_k, can be written in terms of sampled versions of the received signals ($s_i(k)$), the noise sources ($n_i(k)$) and the optimal weights (w_i):

$$y(k) = \left(\sum_{i=1}^{N} w_i s_i(k)\right) + \left(\sum_{i=1}^{N} w_i n_i(k)\right), \quad (7.2)$$

where i is the ith antenna element. The first term is the signal term, and the second term is the noise term. For a planar array receiving a signal from a far-field source, each element sees a phase shifted version of the original signal,

$$s_i(k) = \alpha_i s_0(k) \quad (7.3)$$

$s_0(k)$ is the sampled complex baseband representation of the transmitted signal, and α_i is a complex constant representing the phase shift of the signal and taken to have unit magnitude. Under these conditions, $y(k)$, reduces to

$$y(k) = \left(\sum_{i=1}^{N} w_i \alpha_i s_0(k)\right) + \left(\sum_{i=1}^{N} w_i n_i(k)\right), \quad (7.4)$$

It can be shown from Equation (1.1) that the optimal solution for the weights is $w_i = w_0 \alpha_i^*$, where w_0 is a real constant, Figure 7.4. Substituting the weights into Equation (7.4), the output reduces to

$$y(k) = \left(\sum_{i=1}^{N} w_0 |\alpha_i|^2 s_0(k)\right) + \left(\sum_{i=1}^{N} w_0 \alpha_i^* n_i(k)\right) \quad (7.5)$$

$$= \left(w_0 s_0(k) \sum_{i=1}^{N} 1\right) + \left(w_0 \sum_{i=1}^{N} e^{-\phi_i} n_i(k)\right) \quad (7.6)$$

$$= (w_0 s_0(k) N) + \left(w_0 \sum_{i=1}^{N} e^{-\phi_i} n_i(k)\right) \quad (7.7)$$

User's Signal

Figure 7.5. An example of a planar array adaptive system where the beam-forming is performed before any amplifiers of down-converters (represented by the noise source).

Again, using the previous assumptions for the noise, weights and α_i's, $y(k)$ reduces to

$$y(k) = Nw_0 s_0(k) + n_i(k) \qquad (7.13)$$

The resulting SNR is

$$SNR_{planar,ideal} = \frac{E\left[(Nw_0 s_0(k))(Nw_0 s_0(k))^*\right]}{E\left[(n_i(k))(n_i(k))^*\right]} \qquad (7.14)$$

$$= \frac{N^2 w_0^2 \sigma_s^2}{\sigma_n^2} \qquad (7.15)$$

$$= \frac{N^2 w_0^2 \sigma_s^2}{\sigma_n^2} \qquad (7.16)$$

$$\qquad (7.17)$$

If it is possible to perform the beam-forming (weights) before the down-conversion (noise) then the SNR grows as N^2. In addition, gain in the weights (w_0) can be added without penalty. However, the system in Figure 7.5 is not practical. There are several important physical limitations that are over looked in the model. First of all, the samplers in the system need to operate at twice the carrier frequency,

Figure 7.4. The weight trajectories in the complex plane for a planar array adaptive receiver converging to the optimal weights.

ϕ_i is the phase shift seen by each array element. The SNR for this system is

$$SNR_{planar} = \frac{E\left[(Nw_0 s_0(k))(Nw_0 s_0(k))^*\right]}{E\left[\left(w_0 \sum_{i=1}^N e^{-\phi_i} n_i(k)\right)\left(w_0 \sum_{i=1}^N e^{-\phi_i} n_i(k)\right)^*\right]} \qquad (7.8)$$

$$= \frac{N^2 w_0^2 \sigma_s^2}{w_0^2 \sum_{i=1}^N E\left[(e^{-\phi_i} n_i(k))(e^{-\phi_i} n_i(k))^*\right]} \qquad (7.9)$$

$$= \frac{N^2 \sigma_s^2}{N\sigma_n^2} \qquad (7.10)$$

$$= \frac{N\sigma_s^2}{\sigma_n^2} \qquad (7.11)$$

Since the signals add coherently and the noise adds incoherently the SNR for this system scales linearly with the number of the elements in the array.

The system in Figure 7.3 is the "typical way" in which planar arrays adaptive systems are implemented; however, if the beamforming could be done before the noise is added, then the resulting SNR is greater than showed previously, Figure 7.5.

In this case, the output signal is

$$y(k) = \left(\sum_{i=1}^N w_i s_i(k)\right) + n_i(k) \qquad (7.12)$$

Figure 7.7. An example of a narrowband lens array adaptive receiver shown for the single user case. The received signal, $s(t)$, is focused onto detectors on the focal surface. The detected signals, $\alpha_i N s(t)$, are amplified and down converted (represented by the noise source). The down converted signals plus noise are sampled (represented by switches) with a sampling period T, giving signals, $x_{i,k}$, used by the algorithm. The algorithm uses the training signal, d_k, to adjust the weights, W, such that the SNR at the output, y_k, is maximized.

Just like the pre-noise beam-forming planar array system, the SNR is

$$SNR_{lens,ideal} = \frac{E\left[\left(e^{j\phi}Ns_0(k)\right)\left(e^{j\phi}Ns_0(k)\right)^*\right]}{E\left[(n(k))(n(k))^*\right]} \quad (7.18)$$

$$= \frac{N^2\sigma_s^2}{\sigma_n^2} \quad (7.19)$$

Unfortunately, it is not possible to build an ideal lens array. The array can not focus all of the received signals to a single point, but to a focal spot, Figure 7.7. This means that the signal signal power is distributed over many detectors. For the formulation of the performance of the system in Figure 7.7, let us redefine the α_i's as a measure of how well the lens focuses. Assuming that the lens array elements receive signals from a far-field source of the form,

$$s_i(k) = e^{j\phi_i}s_0(k) \quad (7.20)$$

where ϕ_i describes the phase shift seen by each array element, the signals received by the detectors on the focal surface is given by

$$s_{i,det}(k) = \alpha_i N s_0(k) \quad (7.21)$$

Figure 7.6. An example of a narrowband lens array adaptive receiver shown for the single user case. The received signal, $s(t)$, is perfectly focused onto a detector on the focal surface. The detected signal, $e^{-j\phi}Ns(t)$, are amplified and down converted (represented by the noise source). The down converted signals plus noise are sampled (represented by switches) with a sampling period T. Since the lens perfectly focuses the received signal, the algorithm only chooses which detector to switch to the output.

since the act of down conversion adds noise to the system. Secondly, the beam-forming hardware needs to located very close to the antenna array, since cable loss reduces signal power and adds thermal noise. It is for this reason that LNAs are placed close to the antenna elements and become one of the dominate noise sources.

A lens array, Figure 7.6, focuses the received signal, $s(t)$, to a point on the focal surface. The transmission line delays in the lens and the spatial delays between lens elements and the focal point act as the beam-forming network (weights) shown in Figure 7.5. For an ideal lens, all of the lens array received signals, $s_i(k)$, sum coherently at the focal point, $e^{j\phi}Ns_0(k)$. Here the ϕ indicates that in general, the coherent sum of the signals is out of phase from the original signal. In the ideal lens system, the algorithm decision is to choose which detector is illuminated by the lens.

This is as if the lens array does not focus the signals but illuminate the focal surface equally. In this case, the SNR of Equation (7.28) reduces to

$$SNR_{lens} = \frac{N\sigma_s^2}{\sigma_n^2} \qquad (7.29)$$

which is the same as for the planar array case in Figure 7.3. When all the α_i's are equal to zero except for one which has unit magnitude, then in this case all the beam-forming in done before the noise is added (the ideal lens case). Then Equation (7.28) becomes

$$SNR_{lens} = \frac{N^2\sigma_s^2}{\sigma_n^2} \qquad (7.30)$$

and equals the result for the ideal lens case.

Given the two previous examples, the hybrid lens system should have a better performance than the "typical" planar array system, Figure 7.3, from the partial beam-forming done before the noise is added. Is this true in general? The SNR Equation (7.28) depends on the distribution of the signal among the detectors described by the summation

$$\sum_{i=1}^{N} |\alpha_i|^2 \qquad (7.31)$$

Let us start with the no pre-noise beam-forming†, where $|\alpha_i| = \frac{1}{N} = \alpha_0$. Equation (7.31) becomes

$$\sum_{i=1}^{N} |\alpha_0|^2 \qquad (7.32)$$

Then re-distribute the signal by letting $\alpha_m = \alpha_0 - c$ and $\alpha_n = \alpha_0 + c$ where m, n are the indexes of two detectors. The Equation (7.32) becomes

$$\left(\sum_{i=1, i\neq m,n}^{N} |\alpha_0|^2\right) + (\alpha_0 - c)^2 + (\alpha_0 + c)^2 = \left(\sum_{i=1, i\neq m,n}^{N} |\alpha_0|^2\right) + 2\alpha_0^2 + 2c^2 \qquad (7.33)$$

$$= \left(\sum_{i=1}^{N} |\alpha_0|^2\right) + c^2 \qquad (7.34)$$

$$> \sum_{i=1}^{N} |\alpha_0|^2 \qquad (7.35)$$

The α_i's shows how well the lens coherently focuses the re-radiated signals to each detector and are constrained such that

$$\sum_{i=1}^{M} |\alpha_i| \leq 1 \qquad (7.22)$$

Here the summation is taken from 1 to M for the M detectors on the focal surface. Interestingly, if the detectors are placed on the focal surface in accordance to the 2-dimensional sampling theorem, then $M \approx N$. For this analysis, we will assume $M = N$ so that both the planar array system and lens array system have the same number of degrees of freedom (weights).

The output for this system is given by

$$y(k) = \left(\sum_{i=1}^{N} w_i \alpha_i N s_0(k)\right) + \left(\sum_{i=1}^{N} w_i n_i(k)\right), \qquad (7.23)$$

Again, using $w_i = w_0 \alpha_i^*$, the output becomes

$$y(k) = \left(w_0 N s_0(k) \sum_{i=1}^{N} |\alpha_i|^2\right) + \left(w_0 \sum_{i=1}^{N} \alpha_i^* n_i(k)\right), \qquad (7.24)$$

The SNR for this system is

$$SNR_{lens} = \frac{E\left[\left(w_0 N s_0(k) \sum_{i=1}^{N} |\alpha_i|^2\right)\left(w_0 N s_0(k) \sum_{i=1}^{N} |\alpha_i|^2\right)^*\right]}{E\left[\left(w_0 \sum_{i=1}^{N} \alpha_i^* n_i(k)\right)\left(w_0 \sum_{i=1}^{N} \alpha_i^* n_i(k)\right)^*\right]} \qquad (7.25)$$

$$= \frac{w_0^2 N^2 \sigma_s^2 \left|\sum_{i=1}^{N} |\alpha_i|^2\right|^2}{w_0^2 \sum_{i=1}^{N} E\left[(\alpha_i^* n_i(k))(\alpha_i^* n_i(k))^*\right]} \qquad (7.26)$$

$$= \frac{w_0^2 N^2 \sigma_s^2 \left(\sum_{i=1}^{N} |\alpha_i|^2\right)^2}{w_0^2 \sigma_n^2 \sum_{i=1}^{N} |\alpha_i|^2} \qquad (7.27)$$

$$= \frac{N^2 \sigma_s^2}{\sigma_n^2} \sum_{i=1}^{N} |\alpha_i|^2 \qquad (7.28)$$

The Equation (7.28), describes the SNR for an imperfect lens array system. The system can be considered as a hybrid system with beam-forming before noise is added (lens) and beam-forming after noise is added (weights). When the α_i's have equal magnitude, $|\alpha_i| = \frac{1}{N}$, then beam-forming occurs after the addition of the noise.

Therefore the re-distribution of the signal, pre-noise beam-forming, increases the processing gain of the system. The limiting case is the perfect lens system.

There are several practical issues in adaptive system that still need to be addressed. It is true that the optimal solution for an adaptive system adds in a signal no matter how small its signal power. For a lens system, most of the signal power is concentrated on a few detectors. If a system only processes the detector signals with the most signal power, the resulting SNR will only be slightly less than the optimal SNR,

$$SNR_{lens,M} = \frac{N^2 \sigma_s^2}{\sigma_n^2} \sum_{i=1}^{M} |\alpha_0|^2 \qquad (7.36)$$

$M < N$ and determines the number of detector signals to be used. Practically speaking, a processor operating on a subset of the detector signals will have nearly optimal SNR with far less computation load. In real adaptive systems, the samplers also quantize the detector signals. If these quantizers do not have enough dynamic range (bits), then the quantization noise may obscure the algorithms ability to estimate very weak signals. Finally, there is a fundamental tradeoff between the weight noise (how well the algorithm estimates the optimal weights) and convergence to the optimal weight solution. The faster the algorithm converges to the solution, the more weight noise there is at the output, which reduces the systems SNR. For the lens array system, this tradeoff is different than for the "typical" planar array system and may have some addition practical advantages.

It has been shown that lens array can improve a adaptive systems SNR though pre-noise beam-forming. How this SNR increases as the number of elements in the lens array increase is still unknown. Unfortunately, real lens arrays are lossy,

$$\sum_{i=1}^{N} |\alpha_i| < 1 \qquad (7.37)$$

and the loss increases with lens size,

$$|\alpha_i| \propto N^{-b}, \qquad 0 < b < 1 \qquad (7.38)$$

Figure 7.8: Image pattern for "desired" user at 30° in multi-user model.

This loss is a result of path loss between lens array elements and elements on the focal surface. It does not add any additional noise to the system but does reduce signal power. The affect of this loss on adaptive lens array systems is still unknown.

7.3.3 Adaptive Lens Array Systems with Multiple Users

The multi-user case and the user/jammer case are examples of a "desired" user being interfered by an "other" user. Preliminary simulations for the multi-user case are done with the "other" user at twice the QPSK modulation data rate and equal transmit power moving towards the "desired" user. The user is 20° off axis, Figure 7.8. When the "other" user moves close to the "desired" user as in Figure 7.9, then the desired user's weight adjust to null out the "other" user, Figure 7.10(a) and 7.11. The weight plot, Figure 7.10(b) and the radiation plots, Figure 7.13, for the planar array show similar results as in the lens array case. Cuts in the radiation patterns along $\chi = 0$ show that the lens and planar arrays create the same depth nulls when the "other" user is close to the "desired" user showing the limitation in the array layout, Figure 7.13. The planar array creates deeper nulls when the "other" user is far from the "desired" user to reduce its power below the noise floor since planar

Figure 7.10. The movement of the optimal weight for lens array (a) and planar array (b) adaptive systems as the "other" user moves closer to the "desired" user. The final realization of the optimal weights are the numbered data points.
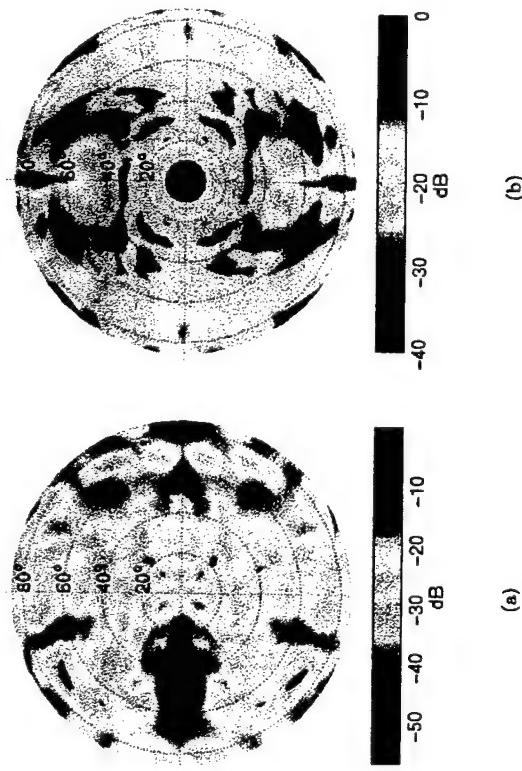
Figure 7.9. Image pattern for "other" user at −40° (a), 0° (b), 20° (c) and 25° (d) in multi-user simulation.

Figure 7.12. The radiation patterns for planar array system in multi-user environment with "desired" user at 30° and the "other" user at −40° (a),0° (b), 20° (c) and 25° (d).
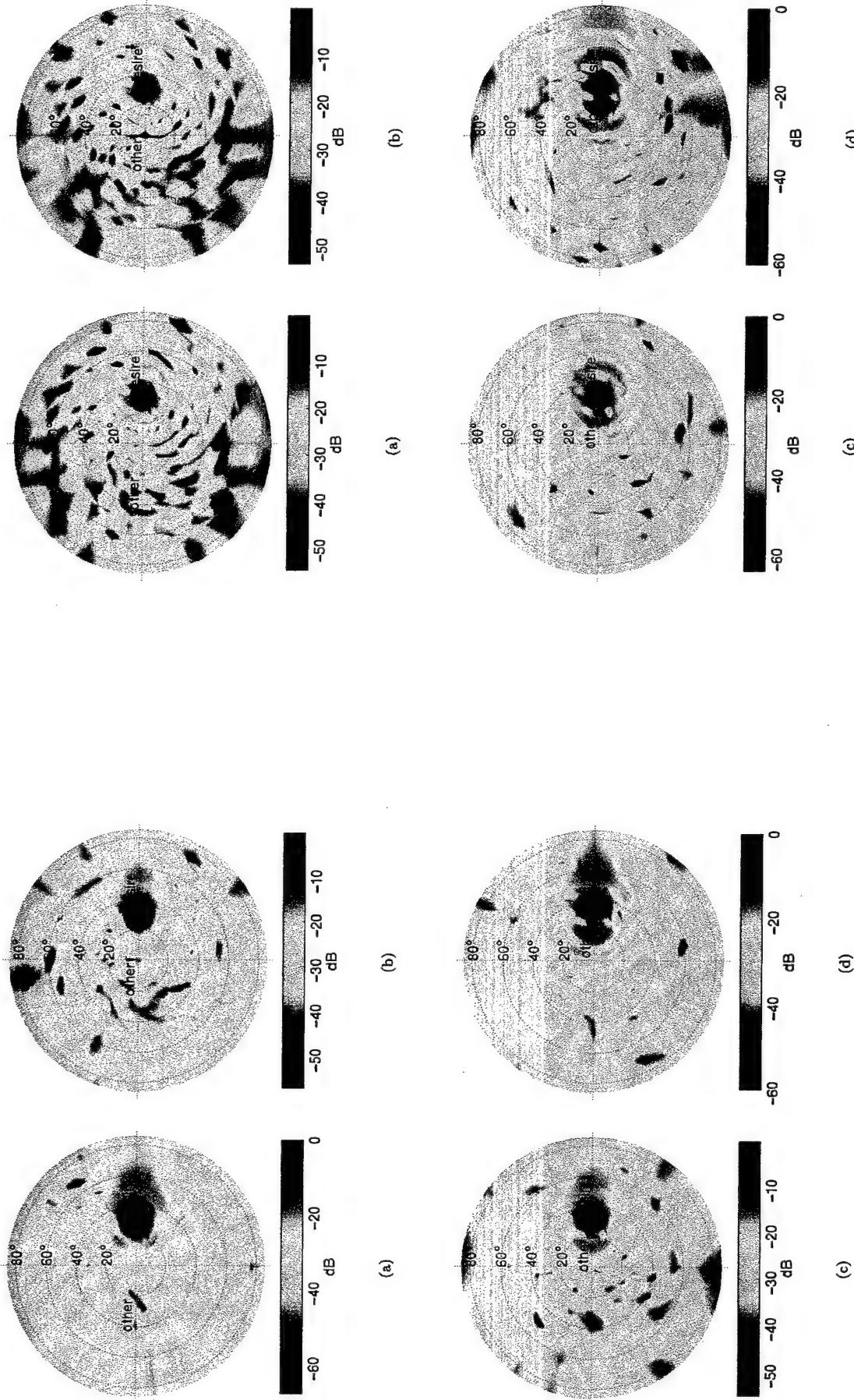
Figure 7.11. The radiation patterns for lens array system in multi-user environment with "desired" user at 30° and the "other" user at −40° (a),0° (b), 20° (c) and 25° (d).

arrays have less loss than lens arrays. These results suggest that lens array and a planar array have the same ability to null out "other" users and that there is not a fundamental limitation in achievable radiation patterns.

7.3.4 Adaptive Lens Array Systems with Multi-path

For the multi-path case, both lens and planar arrays steer beams in the directions of the received signals for the "desired" user, Figure 7.14. The gain in each beam is proportional to the strength of the signal in each path. For example, path 2 has 10 dB less power that path 1, and the difference in there respective array gains is about 10 dB. Again, the lens array system (Figure 7.15) only uses the detectors illuminated by the "desired" user, Figure 7.16(b). Since the signals for the three paths are correlated, they form a standing wave on the surface of the arrays causing the weighs in the planar array form on sets of of circles instead of just one, Figure 7.16(a). The planar arrays still have a better SNR of 23.5 dB, versus the lens arrays of 15.1 dB.

The simulations in section 6.3 are impractical from the point of view that the LMS algorithm knows the transmitted data for the training signal and that the sources are stationary. In a realistic situations, neither of theses conditions are valid. The simulations need to be repeated with decision-directed feedback used for the training signal and with the transmitting sources moving. An important question is whether the weights that are turned "off" will turn back "on" when needed. One simple solution is a second algorithm which uses a larger set of weights and check if new weights are needed. Another solution is to devise a means of estimating the amount of "desired" user signal at each detector to determine which weights should be "on".

A lens array may have an advantage in statistical diversity systems where it naturally decorrelates signals. Previous statistical diversity systems use antennas that are separated by many wavelengths (on the order of 10λ) to decorrelate the signals, [8, 9, 10, 12, 63]. By using a lens array, the same order of diversity can

(a)

(b)
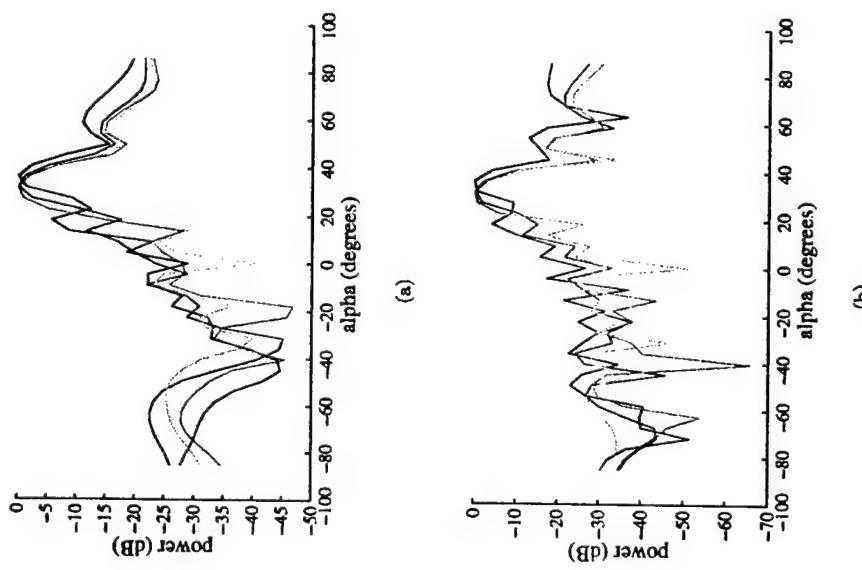
Figure 7.13. $\chi = 0$ cuts for the radiation patterns in figure 7.11 (a) and figure 7.13 (b). The cuts are shown for the "other" user at -40° (yellow), 0° (green), 20° (red) and 25° (blue).

Figure 7.15: The image pattern and detector layout for multi-path environment.

be achieved in a much smaller space. In most statistical diversity systems, the received signals are first decorrelated and then combined. Since a lens array already partially decorrelates the signals, the processor can finish decorrelating the signals with minimal computational power. Turin [63] shows that the optimal combination of the received signals depends on the first- and second-order statistics of the received signals. In a lens array, this is the average of the mean and correlation between the detector signals over all incident angles. If both the mean and covariance of the detector signals are non-zero, then the optimal sum of the signals is a mixture of coherent (for non-zero mean) and non-coherent (for non-zero covariance) detection. From the simulations in section 5.3.2, the phase variations on the signals can be uniformly distributed depending on the lens design. A uniform phase distribution has a non-zero mean and only a non-coherent sum of the signals, which has lower SNR than a

(a)

(b)

Figure 7.14. The radiation patterns for an adaptive lens array (a) and the planar array (b) for three paths of power 0 dB (1), -10 dB (2) and -3 dB (3) in a multi-path environment.

BIBLIOGRAPHY

[1] S. Hollung, **Quasi-optical Transmit/Receive Lens Amplifier Arrays**, Ph.D. thesis, University of Colorado at Boulder, 1998.

[2] D. T. McGrath, "Planar three-dimensional constrained lens," **IEEE Transactions on Antenna and Propagation**, vol. 34, Jan. 1986.

[3] J. G. Proakis, **Digital Communications**, McGraw-Hill, 3rd edition, 1995.

[4] L. W. Couch II, **Digital and Communication Systems**, Macmillan Publishing, 4th edition, 1993.

[5] E. A. Lee and D. G. Messerschmitt, **Digital Communication**, Kluwer Academic Publisher, 1st edition, 1994.

[6] R. Klemm, **Space-Time Adaptive Processing**, Institution of Electrical Engineers, 1st edition, 1998.

[7] R. Nitzberg, **Radar Signal Processing and Adaptive Systems**, Artech House, 1st edition, 1999.

[8] D. G. Brennan, "Linear diversity combining techniques," **Proceedings of IRE**, vol. 47, pp. 1075–1102, 1959.

[9] D. G. Brennan, "On the maximum signal-to noise ratio realizable from several noisy signals," **Proceedings of IRE**, vol. 43, pp. 1530, Oct. 1953.

[10] J. N. Pierce, "Theoretical diversity improvement in frequency-shift keying," **Proceedings of IRE**, vol. 46, pp. 903–910, May 1958.

[11] G. L. Turin, "Communication through noisy, random-multipath channels," **IRE Convention Record**, pp. 154–166, 1956.

[12] Price and Green, "A communication technique for multipath channels," **Proceedings of IRE**, vol. 46, pp. ?, Mar. 1958.

[13] P. Howells, "Intermediate frequency side-lobe canceller," **U. S. Patent 3,202,990**, Aug. 1965.

[14] B. Widrow and J. McCool, "A comparison of adaptive algorithms based on the methods of steepest descent and random search," **IEEE Transactions on Antenna and Propagation**, vol. 24, pp. 615–637, Sept. 1976.

[15] L. J. Griffiths, "A simple adaptive algorithm for real-time processing in antenna arrays," **Proceedings of IEEE**, pp. 1696–1704, Oct. 1969.

[16] R. O. schmidt, "Multiple emitter location and signal parameter estimation," **IEEE Transactions on Antenna and Propagation**, pp. 276–280, Mar. 1986.

[17] R. Kumaresan and D. W. Tufts, "Estimating the angle of arrival of multiple plane waves," **IEEE transactions on Aerospac Electronic Systems**, pp. 134–139, Jan. 1983.

[18] B. Widrow, P. E. Mantey, L. J Griffiths, and B. B. Goode, "Adaptive antenna systems," **Proceedings of IEEE**, vol. 12, pp. 2143–2159, Dec. 1967.

[19] B. Widrow and S. D. Stearns, **Adaptive Signal Processing**, Prentice-Hall, 1st edition, 1985.

[20] G. Kriehn, A. Kirulata, P. E. X. Silveira, S. Weaver, S. Kraut, K. Wagner, R. T. Weverka, and L. Griffiths, "Optical beamtap beam-forming and jammer-nulling

system for broadband phase-array antennas," **Applied Optics**, pp. 212–230, Jan. 2000.

[21] R. C. Hansen, **Phased Array Antennas**, John Wiley and Sons, 1st edition, 1998.

[22] J. L. Butler and R. Lowe, "Beam forming matrix simplifies design of electronically scanned antennas," **Electronic Design**, vol. 9, pp. 170–173, 1961.

[23] J. N. Nolen, **Synthesis of Multiple Beam Networks for Arbitrary Illuminations**, Ph.D. thesis, 1965.

[24] W. D. White, "Pattern limitations in multiple-beam antennas," **IEEE Transactions on Antenna and Propagation**, vol. 18, pp. 430–436, July 1962.

[25] J. L. Allen, "A theorical limitation on the formation of lossless multiple beams in linear arrays," **IEEE Transactions on Antenna and Propagation**, vol. 9, pp. 350–352, July 1961.

[26] J. P. Shelton, "Multibeam planar arrays," **Proceedings og IEEE**, vol. 56, pp. 430–436, Nov. 1968.

[27] R. A. York and Z. B. Popović, **Active and Quasi-Optical Arrays for Solid-State Power Combining**, John Wiley and Sons, 1st edition, 1997.

[28] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, **Feedback Control of Dynamic Systems**, Addison-Wesley Publishing Company, 3rd edition, 1994.

[29] Elliott R. Brown, "Rf-mems switches for reconfigurable integrated circuits," **IEEE Transtransactons on Microwave Theory Techniques**, vol. 46, pp. 1868–1880, 1998 Nov.

[30] A.M. Johnson and D.H. Austin, "Microwave switching by picosecond photoconductivity," **IEEE Quantum Electronics**, June 1975.

[31] **Millimeter-Wave Switching by Optically Generated Plasma in Silcon**, vol. 14, Nov. 1978.

[32] P.R. Herczfeld, A. S. Daryoush, P.Stabile A. Rosen, , and V.M. Contarino, "Optically controlled microwave devices and circuits," **RCA Review**, p. 528, Dec. 1985.

[33] P. J. Stabile, A. Rosen, and P. R. Herczfeld, "Optically controlled lateral pin diodes and microwave control circuits," **RCA Review**, pp. 443–456, Dec. 1986.

[34] P. Stabile Rosen, W. Janton, P. Basile A. Gombar, J. Delmaser, and R. Hurwitz, "Laser-activated pin diode switch for rf applications," **IEEE Transtransactons on Microwave Theory Techniques**, vol. 37, pp. 1255–1257, Aug. 1989.

[35] A. Guenther, M. Kristian, and T. Martin, **Opening Switches**, Plenum, 1987.

[36] Rosen, P. J. Stabile, A. M. Gomar, A. M. Gombar, W. M. Janton, A. Bahasafri, and P. Hercfeld, "100 kw dc-biased, all semiconductor switch using si pin diodes and algaas 2-d laser arrays," **IEEE Photonics Technology Letters**, pp. 132–134, June 1989.

[37] J.L. Freeman, S. Ray, D.L. West, A. G. Thompson, and M. J. LaGasse, "Microwave control using a high-gain bias-free optoelectronic switch," **Optical Technology for Microwave Applications SPIE**, vol. 5, pp. 320–325, 1991.

[38] S. S. Gevorgian, "Short-circuit photocurrent-controlled microwave pin diode switch," **Microwave and optical Technology Letters**, vol. 7, no. 12, pp. 553–555, Aug. 1994.

[39] B. G. Streetman, **Solid State Electronic Devices**, Prentice Hall, 3rd edition, 1990.

[40] A. Yariv, **Optical Electronics**, Saunders College, 4th edition, 1991.

188

[51] W. Rotman and R. F. Turner, "Wide-angle microwave lens for line source applications," **IEEE Transactions on Antenna and Propagation**, vol. 11, pp. 623–632, Nov. 1963.

[52] W. L. Stutzman and G. A. Thiele, **Antenna Theory and Design**, John Wiley and Sons, 1st edition, 1981.

[53] T. Kailath, **Linear Systems**, Prentice-Hall, 2nd edition, 1997.

[54] Goodman, **Introduction to Fourier Optics**, McGraw-Hill, 2nd edition, 1996.

[55] B. Widrow, J. McCool, J. R. Glover jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong jr., and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," **Proceedings of IEEE**, vol. 63, pp. 1692–1716, Dec. 1975.

[56] D. E. Dudgeon and R. M. Mersereau, **Multidimensional Digital Signal Processing**, Prentice-Hall, 1st edition, 1984.

[57] B. Widrow and E. Walach, "On the statistical efficiency of the lms algorithm with nonstationary inputs," **IEEE Transactions on Information Theory**, pp. 211–221, Mar. 1984.

[58] D. C. McLernon, "Analysis of lms algorithm with inputs from cyclostationary random processes," **Electronics Letters**, pp. 136–138, Jan. 1991.

[59] D. H. Hartman and G. R. Lalk, "Radiant cured optical waveguides on printed circuit boards for photonic interconnection use," **Applied Optics**, p. 40, Jan. 1989.

[60] L. D. Bamford, P. S. Hall, and A. Fray, "Calculation of antenna mutual coupling form far radiated fields," **Electronic Letters**, pp. 1299–1301, July 1993.

187

[41] M. L. VanBlaricum, "Photonic antenna reconfiguration: A status survey," **Proceedings of the SPIE, Photonics and Radio Frequency II**, pp. 180–189, 1998 July.

[42] A. S. Nagra, O. Jerphagnon, P. Chavarka, M. L. VanBlaricum, and R. A. York, "Bias free optical control of microwave circuits and antenna using improved optically variable capacitors," in **Internation Microwave Symposium Digest**, June 2000, pp. 687–690.

[43] P. R. Gray and R. G. Meyer, **Analysis and Design of Analog Integrated Circuits**, John Wiley and Sons, 3rd edition, 1993.

[44] T. Marshall, M. Forman, and Z. Popovic, "Two ka-band quasi-optical amplifier arrays," **IEEE Transtransactons on Microwave Theory Techniques**, Dec. 1999.

[45] C. A. Balanis, **Antenna Theory Analysis and Design**, John Wiley and Sons, 2rd edition, 1997.

[46] C. C. Lin, "A microwave coupler throught a narrow slot in the common groung plane of a two-sided microstrip circuit," **Microwave and Optical Technology Letters**, pp. 543–547, Nov. 1991.

[47] J. Brown, "Microwave lenes," 1953.

[48] H.Gent, "Bootlace aerial," **Royal Radar Establishment Journal**, pp. 47–57, Oct. 1957.

[49] J. B. L. Roa, "Multifocal three-dimensional bootlace lenses," **IEEE Transactions on Antenna and Propagation**, vol. 30, pp. 1050–1056, 1982.

[50] C. M. Rappaport and A. I. Zaghloul, "Optimized three-dimmensional for wide-angle scanning," **IEEE Transactions on Antenna and Propagation**, vol. 33, no. 11, pp. 1227–1236, Nov. 1985.

[61] J. A. G. Malherbe, "Analysis fo a linear antenna array including the effects of mutual coupling," **IEEE Transactions on Education**, pp. 29–34, Feb. 1989.

[62] D. E. J. Humphrey and V. F. Fusco, "A mutual coupling model for microstrip patch antenna pairs with arbitrary orientation," **Microwave and Optical Technology Letters**, pp. 230–233, June 1998.

[63] G. L. Turin, "On optimal diversity reception," **IRE Transactions on Information Theory**, pp. 154–166, July 1961.

[64] S. N. Diggavi, J. J. Shynk, and A. J. Laub, "Directional-of-arrival estimation for a lens-base array," **IEEE Transactions on Antenna and Propagation**, pp. 666–675, May 1994.

APPENDIX A

MATLAB CODE

This appendix contains all the *MatLab* code used in the numerical modeling of the lens arrays and LMS adaptive systems.

Numerical Model Main Code

——————— antenna_pattern.m ———————

```
function [antenna_ptrn]=antenna_pattern(antenna,alpha_range, Ki_range)

%this function calculates the antenna pattern in effective length

alpha=[alpha_range(1):(alpha_range(2)-alpha_range(1))/(alpha_range(3)-1):...
    alpha_range(2)];
Ki=[Ki_range(1):(Ki_range(2)-Ki_range(1))/(Ki_range(3)-1):Ki_range(2)];

for k=1:length(alpha)
    for l=1:length(Ki)
        [le(k,1),ro(k,1)]=feval(antenna.name,alpha(k),Ki(1),antenna.polar);
    end
end

antenna_ptrn.le=le;
antenna_ptrn.polar=ro;
antenna_ptrn.alpha=alpha;
antenna_ptrn.ki=Ki;
antenna_ptrn.create='antenna_pattern';
```

——————— arrayTF.m ———————

```
function [array_TF]=arrayTF(array,channel,optional)

%this function creates a tranfer function for channel to array
```

```
% the channel inputs are all unity
%optional.polar_source use polarization for source coupling
%                       default is 'yes'
%optional.polar_feed use polarization for feed coupling
%                    default is 'yes'
%optional.pathloss_source use pathloss for source coupling
%                         default is 'yes'
%optional.pathloss_feed use pathloss for feed coupling
%                       default is 'yes'

optional_focus.polar_source='yes';
optional_focus.polar_feed='yes';
optional_focus.pathloss_source='yes';
optional_focus.pathloss_feed='yes';

if nargin == 3
  if isfield(optional,'polar_source')
    optional_focus.polar_source=optional.polar_source;
  end
  if isfield(optional,'polar_feed')
    optional_focus.polar_feed=optional.polar_feed;
  end
  if isfield(optional,'pathloss_source')
    optional_focus.pathloss_source=optional.pathloss_source;
  end
  if isfield(optional,'pathloss_feed')
    optional_focus.pathloss_feed=optional.pathloss_feed;
  end
end

for k=1:length(channel)
  source=channel{k};
  [array_TF(:,k)]=array_focus(array,source,optional_focus);
end
```

_____ array_flux.m _____

```
function [flux]=array_flux(array,I_in,alpha_range,ki_range)

%this function calculates the total radiated flux from an array whose
%elements have input currents I_in
%alpha_range is the range of alpha angles [alpha_min,alpha_max,alpha_N]
%Ki_range is the range of Ki angles [Ki_min,Ki_max,Ki_N]
```

```
alpha=[alpha_range(1):(alpha_range(2)-alpha_range(1))/(alpha_range(3)-1):...
  alpha_range(2)];
Ki=[Ki_range(1):(Ki_range(2)-Ki_range(1))/(Ki_range(3)-1):Ki_range(2)];

flux.alpha=alpha;
flux.Ki=Ki;

for k=1:alpha_range(3)
  comments(['Radiation Pattern ',num2str(k/alpha_range(3),2)]);
  for l=1:Ki_range(3)
    if alpha(k)==0 & l~=1
      flux.E_rad(k,l)=flux.E_rad(1,1);
      flux.P_rad(k,l)=flux.P_rad(1,1);
    else
      a_rad=[-1*sin(alpha(k))*cos(Ki(1));-1*sin(alpha(k))*sin(Ki(1));...
        -1*cos(alpha(k))];

      E_rad_Kl=[0;0;0];
      for m=1:array.N_c
        delay_Klm=-dot(a_rad,array.cell.nfd.pos{m});

        [antenna_m]=pull_antenna(array,m);
        receive.pos=antenna_m.pos+a_rad;
        [theta_tran,phi_tran]=get_direction(antenna_m,receive);

        [le_klm,ro_klm]=feval(antenna_m.name,theta_tran,phi_tran,...
          antenna_m.polar);
        E_rad_Klm=377*((-j*2*pi)/(4*pi))*le_klm*exp(-j*2*pi*delay_Klm)*...
          ro_klm;
        E_rad_Kl=E_rad_Kl+E_rad_Klm;
      end

      flux.E_rad(k,l)=E_rad_Kl;
      flux.P_rad(k,l)=E_rad_Kl'*E_rad_Kl/377;
    end
  end
end

flux.total=0;
alpha_step=abs(alpha(2)-alpha(1));
Ki_step=abs(Ki(2)-Ki(1));

for k=1:alpha_range(3)-1
  for l=1:Ki_range(3)-1
    E_rad_temp=(flux.E_rad(k,l)+flux.E_rad(k+1,l)+flux.E_rad(k,l+1)+flux.E_rad(k,l+1)+...
      flux.E_rad(k+1,l+1))/4;
    pwr_avg=E_rad_temp'*E_rad_temp/377;
    area=sin((alpha(k)+alpha(k+1))/2)*alpha_step*ki_step;
    flux.total=flux.total+pwr_avg*area;
  end
end
```

```
end

_____ directivity.m _____

function [direct,flux]=directivity(rad_ptrn,optional)

%this function calculates the directivity of  a radiation pattern
%it assumes that all points not taken in the pattern have zero power
%flux is the result of the power flux integral
%optional.freq tells which 'measured radiation' pattern to use
switch (rad_ptrn.create)
case('plot_radiation')
    disp('Please use output of rad_pattern')

case('rad_pattern','antenna_pattern','image_pattern')
    flux=0;
    switch (rad_ptrn.create)
        case 'rad_pattern'
            Voc=rad_ptrn.Voc;
        case 'image_pattern'
            Voc=rad_ptrn.Voc;
            rad_ptrn.alpha=rad_ptrn.theta;
            rad_ptrn.Ki=rad_ptrn.phi;
        case 'antenna_pattern'
            Voc=rad_ptrn.le;
    end

    flux=0;
    alpha=rad_ptrn.alpha;
    Ki=rad_ptrn.Ki;
    alpha_step=abs(alpha(2)-alpha(1));
    Ki_step=abs(Ki(2)-Ki(1));

    for k=1:length(rad_ptrn.alpha)-1
        for l=1:length(rad_ptrn.Ki)-1
            Voc_temp=[Voc(k,1),Voc(k+1,1),Voc(k,1+1),Voc(k+1,1+1)];
            pwr_avg=Voc_temp*Voc_temp'/4;
            area=sin((alpha(k)+alpha(k+1))/2)*alpha_step*Ki_step;
            flux=flux+pwr_avg*area;
        end
    end

case('measured')
    flux=0;
    pwr=rad_ptrn.pwr;
    alpha=rad_ptrn.alpha;
    Ki=rad_ptrn.Ki;
```

```
    alpha_step=abs(alpha(2)-alpha(1));
    Ki_step=abs(Ki(2)-Ki(1));

    for k=1:length(rad_ptrn.alpha)-1
        for l=1:length(rad_ptrn.Ki)-1
            pwr_avg=sum([pwr(k,1),pwr(k+1,1),pwr(k,1+1),pwr(k+1,1+1)])/4;
            area=sin((alpha(k)+alpha(k+1))/2)*alpha_step*Ki_step;
            flux=flux+pwr_avg*area;
        end
    end

    disp('not tested yet need a function to  strip out  a rad_ptrn from data')

end

direct=(4*pi*max(max(abs(Voc)))^2)/flux;
if strcmp(rad_ptrn.create,'antenna_pattern')
    flux=flux*((2*pi*377)^2)/((4*pi)^2*377);
end

_____ get.first.lobe.m _____

function [first_lobe]=get_first_lobe(ptrn,optional)

%this function calculates the peak power of the first side lobe  of
%the a pattern in eigth direction.
%optional.center

if strcmp(ptrn.create,'rad_pattern') | strcmp(ptrn.create,'image_pattern')
    pwr=abs(ptrn.Voc).^2;
    if strcmp(ptrn.create,'rad_pattern')
        ptrn.theta=ptrn.alpha;
        ptrn.phi=ptrn.Ki;
    end

    if exist('optional')
        HPBW.center=optional.center;
    else
        [K_list,K_index]=max(pwr);
        [pwr_max,1_index]=max(k_list);
        K_index=K_index(1_index);
        HPBW.center=[ptrn.theta(k_index),ptrn.phi(1_index)];
    end

    x_max=HPBW.center(1)*cos(HPBW.center(2));
    y_max=HPBW.center(1)*sin(HPBW.center(2));
    x_y_points=j*ones(2,8);
```

```
if x_y_points(1,m)~=j
    theta=sqrt(x_y_points(1,m)^2+x_y_points(2,m)^2);
    if theta>.001
        phi=atan2(x_y_points(2,m),x_y_points(1,m));
        if phi < 0
            phi=phi+2*pi;
        end
    else
        phi=0;
    end

    vector2=[sin(theta)*cos(phi);sin(theta)*sin(phi);cos(theta)];
    first_lobe.angles(m)=acos(vector1'*vector2);
    else
        count=count-1;
        first_lobe.angles(m)=0;
    end
    end

    first_lobe.angle_avg=sum(first_lobe.angles)/count;
    first_lobe.pwr_avg=sum(first_lobe.pwr)/count;

else
    disp('use output of rad_pattern or image_pattern')
end
```

_____ get_HPBW.m _____

```
function [HPBW]=get_HPBW(ptrn,optional)

%this function calculates the HPBW of the a pattern in eigth direction.
%the average angel is the angel from the center to the outward
%optional.center

if strcmp(ptrn.create,'rad_pattern') | strcmp(ptrn.create,'image_pattern')

Voc=ptrn.Voc;
if strcmp(ptrn.create,'rad_pattern')
    ptrn.theta=ptrn.alpha;
    ptrn.phi=ptrn.ki;
end

[X,Y]=meshgrid(ptrn.phi,ptrn.theta);

if exist('optional')
    HPBW.center=optional.center;
    [Voc_max]=interp2(X,Y,Voc,HPBW.center(2),HPBW.center(1));
```

```
for m=0:7

r_temp=[0:1/40:2];
x_temp=cos(m*pi/4)*r_temp+x_max*ones(size(r_temp));
y_temp=sin(m*pi/4)*r_temp+y_max*ones(size(r_temp));

for k=1:length(r_temp)
    theta(k)=sqrt(x_temp(k)^2+y_temp(k)^2);
    if theta>.001
        phi(k)=atan2(y_temp(k),x_temp(k));
        if phi(k) <0
            phi(k)=phi(k)+2*pi;
        end
    else
        phi(k)=0;
    end
end

[X,Y]=meshgrid(ptrn.phi,ptrn.theta);
[r_intp]=interp2(X,Y,pwr,phi,theta);

first=0;
second=0;
for k=1:length(r_intp)-1
    if r_intp(k) <= r_intp(k+1) & first==0
        first=1;
        rise=k;
    end

    if r_intp(k) > r_intp(k+1) & first==1 & second==0
        second=1;
        fall=k+1;
    end

end

if first==1 & second==1
    [first_lobe.pwr(m+1),index]=max(r_intp([rise:fall]));
    r_lobe=r_temp(rise+index-1);
else
    r_lobe=0;
end

x_y_points(1,m+1)=r_lobe*cos(m*pi/4)+x_max;
x_y_points(2,m+1)=r_lobe*sin(m*pi/4)+y_max;
end

vector1=[sin(HPBW.center(1))*cos(HPBW.center(2));...
    sin(HPBW.center(1))*sin(HPBW.center(2));...
    cos(HPBW.center(1))];

count=8;
for m=1:8
```

```
  pwr_max=Voc_max*Voc_max';
  HPBW.pwr_max=pwr_max;
else
  [k_list,k_index]=max(Voc);
  [Voc_max,l_index]=max(k_list);
  k_index=k_index(l_index);
  HPBW.center=[ptrn.theta(k_index),ptrn.phi(l_index)];
  pwr_max=Voc_max*Voc_max';
  HPBW.pwr_max=pwr_max;
end

x_max=HPBW.center(1)*cos(HPBW.center(2));
y_max=HPBW.center(1)*sin(HPBW.center(2));
x_y_points=j*ones(2,8);
for m=0:7

r_temp=[0:1/40:1];
x_temp=cos(m*pi/4)*r_temp+x_max*ones(size(r_temp));
y_temp=sin(m*pi/4)*r_temp+y_max*ones(size(r_temp));

for k=1:length(r_temp)
  theta(k)=sqrt(x_temp(k)^2+y_temp(k)^2);
  if theta(k)>.001
    phi(k)=atan2(y_temp(k),x_temp(k));
    if phi(k)<0
      phi(k)=phi(k)+2*pi;
    end
  else
    phi(k)=0;
  end
end

[Voc_intp]=interp2(X,Y,Voc,phi,theta);
r_intp=abs(Voc_intp).^2;

first=0;
for k=1:length(r_intp)-1
  if r_intp(k) >= pwr_max/2 & r_intp(k+1)<=pwr_max/2 & first==0
    first=1;
    slope=(r_temp(k+1)-r_temp(k))/(r_intp(k+1)-r_intp(k));
    intercept=r_temp(k)-slope*r_intp(k);
    r_3dB=slope*(pwr_max/2)+intercept;

    slope=(Voc_intp(k+1)-Voc_intp(k))/(r_temp(k+1)-r_temp(k));
    intercept=Voc_intp(k) -slope*r_temp(k);
    Voc_3dB=slope*r_3dB+intercept;
    HPBW.phase(m+1)=angle(Voc_3dB)-angle(Voc_max);

    x_y_points(1,m+1)=r_3dB*cos(m*pi/4)+x_max;
    x_y_points(2,m+1)=r_3dB*sin(m*pi/4)+y_max;
  end
end
```

```
    end
  end
  vector1=[sin(HPBW.center(1))*cos(HPBW.center(2));...
           sin(HPBW.center(1))*sin(HPBW.center(2));...
           cos(HPBW.center(1))];
  count=8;
  for m=1:8
    if x_y_points(1,m)~=j
      theta=sqrt(x_y_points(1,m)^2+x_y_points(2,m)^2);
      if theta>.001
        phi=atan2(x_y_points(2,m),x_y_points(1,m));
        if phi<0
          phi=phi+2*pi;
        end
      else
        phi=0;
      end
      vector2=[sin(theta)*cos(phi);sin(theta)*sin(phi);cos(theta)];
      HPBW.angles(m)=acos(vector1'*vector2);
    else
      count=count-1;
      HPBW.angles(m)=0;
    end
  end
  if count==0
    HPBW.avg=0;
  else
    HPBW.avg=sum(HPBW.angles)/count;
  end
else
  disp('use output of rad_pattern or image_pattern')
end
```

image_pattern.m

```
function [image_ptrn,mutual_ptrn]=image_pattern(lens,channel,theta_range,phi_range,opt

%this function outputs the det_antenna Voc for an omni direction antenna
%polarized to the center element of the lens feed side for the 'channel'
%input over the specified ranges.
%mutual_prtn is the image produced by the mutual coupling currents
%theta_range is a vector [theta_start,theta_end,theta_N]
%phi_range is a vector [phi_start,phi_end,phi_N]
%optional.polar_source use polarization for source coupling
```

```
%                    default is 'yes'
%optional.polar_feed use polarization for feed coupling
%                    default is 'yes'
%optional.pathloss_source use pathloss for source coupling
%                    default is 'yes'
%optional.pathloss_feed ('yes','no','focal') use pathloss for feed coupling
%                    'focal' uses the focal distance on the optical axis
%                    as the path loss for all antenna elements
%                    default is 'yes'
%optional.surface ('focal',radius) to use the focal surface or a surface of
%                    radius 'radius' default is 'focal'
%optional.det_antenna allows you to define your own detector antenna element
%optional.mutual.use ('yes', 'no','max') calculates the image do to mutual
%                    coupling.  'yes' uses the estimated magnitude and phase
%                    of the coupled currents and 'max' uses just the magnitude
%                    default is 'no'
%optional.mutual.extra is an open field for use by name_mutual.m mutual
%coupling calculations functions

mutual.use='no';
surface='focal';
focal='no';

optional_focus.polar_source='yes';
optional_focus.polar_feed='yes';
optional_focus.pathloss_source='yes';
optional_focus.pathloss_feed='yes';

%initalize default det_antenna
middle=round(lens.N_c/2);
[fd_antenna]=pull_antenna(lens,middle,'feed');
[le_nfd,ro_fd]=feval(fd_antenna.name,0,0,fd_antenna.polar);
J_sph_fd=xyz2sph(0,0);
J_gfd=fd_antenna.local;
ro_source=J_gfd'*J_sph_fd'*ro_fd;
det_antenna.polar=ro_source;
det_antenna.theta_r=0;
det_antenna.name='omni_antenna';
%

if nargin ==5
  if isfield(optional,'polar_source')
    optional_focus.polar_source=optional.polar_source;
  end
  if isfield(optional,'polar_feed')
    optional_focus.polar_feed=optional.polar_feed;
  end
  if isfield(optional,'pathloss_source')
    optional_focus.pathloss_source=optional.pathloss_source;
  end
  if isfield(optional,'pathloss_feed')
    if strcmp(optional.pathloss_feed,'focal')
      optional_focus.pathloss_feed='no';
    else
      optional_focus.pathloss_feed=optional.pathloss_feed;
      focal='no';
    end
  end
  if isfield(optional,'mutual')
    mutual=optional.mutual;
  end
  if isfield(optional,'surface')
    surface=optional.surface;
  end
  if isfield(optional,'det_antenna')
    det_antenna=optional.det_antenna;
  end
end

theta=[theta_range(1):(theta_range(2)-theta_range(1))/(theta_range(3)-1):...
      theta_range(2)];
phi=[phi_range(1):(phi_range(2)-phi_range(1))/(phi_range(3)-1):phi_range(2)];

lin_fd=0;
for k=1:theta_range(3)
  for l=1:phi_range(3)
    comments(['Calculating Image ',num2str(k/theta_range(3),2)]);
    if theta(k)==0 & l~=1
      Voc(k,l)=Voc(1,1);
    else
      if strcmp(surface,'focal')
        [rad]=get_focal(lens, theta(k));
      else
        rad=surface;
      end
      pos=[rad*sin(theta(k))*cos(phi(1));rad*sin(theta(k))*sin(phi(1));...
          rad*cos(theta(k))];
      [detector_k1]=build_det(det_antenna.name,pos,...
          det_antenna.theta_r,1,1,0.5,0.5,0,det_antenna.polar);

      Voc_temp=0;
      for m=1:length(channel)
        source_m=channel(m);

        [Voc_temp2,lin_fd,m]=lens_focus(lens,source_m,detector_k1,...
            optional_focus);

        Voc_temp=Voc_temp+Voc_temp2;

        if k==1 & l==1
          switch(mutual.use)
```

```
        [Voc(k,1)]=mutual_focus(lens,Iin,Iin_fd_mutual,detector_kl,...
          optional_focus);
        end
      end
    end

if strcmp(focal,'yes')
  mutual_ptrn.Voc=Voc/get_focal(lens,0);
else
  mutual_ptrn.Voc=Voc;
end
mutual_ptrn.theta=theta;
mutual_ptrn.phi=phi;
mutual_ptrn.create='image_pattern';

end

_____local_unit_vectors.m_____

function local_unit_vectors(device,optional)

%local_unit_vectors(device,optional)
%this function plots local unit vectors for antennas in the device
%(lens or detectors)
%optiona.fig_num is optional and default is 4
%optional.font_size is optional and allows the letter font size to be changed
%          default is 10
%optional.length length of unit vectors
%          default is 0.25

fig_num=4;
optional_draw.font_size=10;
if nargin==2
  if isfield(optional,'fig_num')
    fig_num=optional.fig_num;
  end
  if isfield(optional,'font_size')
    optional_draw.font_size=optional.font_size;
  end
  if isfield(optional,'length')
    optional_draw.length=optional.length;
  end
end

figure(fig_num)

if findstr(class(device),'cell')
  for k=1:length(device)
```

```
        case {'yes','max'}
            Iin_fd=Iin_fd+Iin_fd_m;
        end
        Voc(k,1)=Voc_temp;
      end
    end
    if strcmp(focal,'yes')
      image_ptrn.Voc=Voc/get_focal(lens,0);
    else
      image_ptrn.Voc=Voc;
    end
    image_ptrn.theta=theta;
    image_ptrn.phi=phi;
    image_ptrn.create='image_pattern';

if strcmp(mutual.use,'yes') | strcmp(mutual.use,'max')

Voc=[];
[mutual_couple]=get_mutual_Z(lens,optional.mutual);
switch(mutual.use)
  case{'yes'}
    Iin_fd_mutual=mutual_couple*Iin_fd;
  case{'max'}
    Iin_fd_mutual=abs(mutual_couple)*abs(Iin_fd);
end

for k=1:theta_range(3)
  for l=1:phi_range(3)

comments([['Calculating Mutual ',num2str(k/theta_range(3),2)]]);

if theta(k)==0 & l~=1
  Voc(k,1)=Voc(1,1);
else
  if strcmp(surface,'focal')
    [rad]=get_focal(lens, theta(k));
  else
    rad=surface;
  end

pos=[rad*sin(theta(k))*cos(phi(1));rad*sin(theta(k))*sin(phi(1));...
  rad*cos(theta(k))];
a_z=(-1*pos)/sqrt(pos'*pos);

[detector_kl]=build_det(det_antenna.name,pos,...
  det_antenna.theta_r,1,0.5,0.5,0,det_antenna.polar);
```

```
device_K=device(k);

    switch device_K.device
    case('detector')
        for m=1:device.N_c
            drawunit(device_k.cell.local{m},device_k.cell.pos{m},'c',....
                optional_draw)
        end
    case('source')
        switch device_k.type
        case('nearfield')
            drawunit(device_K.local,device_k.pos,'m',optional_draw)
        case('farfield')
            drawunit(device_K.local,device_k.pos,'m',optional_draw)
        end
    end
else
    switch device.device
    case('lens')
        for m=1:device.N_c
            drawunit(device.cell.nfd.local{m},device.cell.nfd.pos{m},'g',....
                optional_draw)
        end
    case('detector')
        for m=1:device.N_c
            drawunit(device.cell.local{m},device.cell.pos{m},'c',optional_draw)
        end
    case('source')
        switch device.type
        case('nearfield')
            drawunit(device.local,device.pos,'m',optional_draw)
        case('farfield')
            drawunit(device.local,device.pos,'m',optional_draw)
        end
    end
end
```

────────── make.channel.m ──────────

```
function [channel]=make_channel(alpha_max,alpha_sample)
```

```
%this function uniformly places far field sources over a surfcae from 0
%to alpha_max

alpha=[0:alpha_max/(alpha_sample-0.5):alpha_max]
alpha_step=alpha(2)-alpha(1);

channel_num=1;
channel(channel_num)=build_far(0,0,pi/2,1,[0;1;0]);
for k=2:length(alpha)
    Ki_sample=floor(sin(alpha(k))*2*pi/alpha_step);
    if floor(k/2)==k/2
        Ki=[0:2*pi/Ki_sample:2*pi-2*pi/Ki_sample];
    else
        Ki=[0*pi/Ki_sample:2*pi/Ki_sample:2*pi-pi/Ki_sample];
    end
    for l=1:Ki_sample
        channel_num=channel_num+1
        channel(channel_num)=build_far(alpha(k),Ki(1),pi/2,1,[0;1;0]);
    end
end
```

────────── make_imager.m ──────────

```
function [imager]=make_imager(lens,detector,theta_max,optional)

%this function places the detectors on the image_ptrnr surface spaced according
%the Nyquist sampling therom. I use a small angle approximation that results
%in a little over sampling for lens smaller than radius=3 lambda
%'detector' is the element used
%( right now only written for single antenna for use in '2D' lens, arrays
% detectors are for '1D' lenses)
%'theta_max' range from 0 to theta_max to place the detectors
%optional.image_ptrn will use image pattern to calculate HPBW for sampling
%       use image with beam on optical axis
%       default is to estimate HPBW from lens size
%optional.sampling_scale   ratio of desired sampling rate to Nyquist sampling rate
%       default is 1
%optional.surface ('focal',num) which focal surface to use focal surface or
%                   a ball of radius 'num'
%       default is focal surface
%optional.fit ('ceil','floor') ceil packs the detectors such the whole angle
%       space is covered with max spacing no greater than Nyquist
%       floor packs the detectors starting at the center with Nyquist
%       spacing until theta_max is reached
%       default is floor
```

```matlab
surface='focal';
sampling_scale=1;
estimate='yes';
fit='floor';

if nargin==4
    if isfield(optional,'sampling_scale')
        sampling_scale=optional.sampling_scale;
    end
    if isfield(optional,'surface')
        surface=optional.surface;
    end
    if isfield(optional,'image_ptrn')
        image_ptrn=optional.image_ptrn;
        estimate='no';
    end
    if isfield(optional,'fit')
        fit=optional.fit;
    end
end

switch(estimate)
    case('yes')
        switch(lens.lensing)
            case('2D')
                switch(lens.shape)
                    case('circle')
                        HPBW=2*asin(.255/lens.dimension);
                    case('rect')
                        HPBW(1)=2*asin(1.4/(lens.dimension(1)*pi));
                        HPBW(2)=2*asin(1.4/(lens.dimension(2)*pi));
                end
            case('1Dx')
                switch(lens.shape)
                    case('circle')
                        disp('warning this code does not work with 1D circle lenses')
                    case('rect')
                        HPBW(1)=2*asin(1.4/(lens.dimension(1)*pi));
                        HPBW(2)=0;
                end
            case('1Dy')
                switch(lens.shape)
                    case('circle')
                        disp('warning this code does not work with 1D circle lenses')
                    case('rect')
                        HPBW(1)=0;
                        HPBW(2)=2*asin(1.4/(lens.dimension(2)*pi));
                end
        end
```

```matlab
    case('no').
        HPBW=get_HPBW(image_ptrn);
        switch(lens.lensing)
            case('2D')
                switch(lens.shape)
                    case('circle')
                        HPBW=2*HPBW.avg;
                    case('rect')
                        HPBW(1)=sum(HPBW.angles([1,5]))/2;
                        HPBW(2)=sum(HPBW.angles([3,7]))/2;
                end
            case('1Dx')
                switch(lens.shape)
                    case('circle')
                        disp('warning this code does not work with 1D circle lenses')
                    case('rect')
                        HPBW(1)=sum(HPBW.angles([1,5]))/2;
                        HPBW(2)=0;
                end
            case('1Dy')
                switch(lens.shape)
                    case('circle')
                        disp('warning this code does not work with 1D circle lenses')
                    case('rect')
                        HPBW(1)=0;
                        HPBW(2)=sum(HPBW.angles([3,7]))/2;
                end
        end
end

switch(lens.lensing)
    case('2D')
        switch(lens.shape)
            case('circle')
                sample=HPBW*1.13*0.5*[1 1 ;sqrt(3) -sqrt(3)];
            case('rect')
                sample(:,1)=sampling_scale*HPBW(1)*1.12*[1;0];
                sample(:,2)=sampling_scale*HPBW(2)*1.12*[0;1];
        end
    case('1Dx')
        switch(lens.shape)
            case('circle')
                disp('warning this code does not work with 1D circle lenses')
            case('rect')
                sample(:,1)=sampling_scale*HPBW(1)*1.12*[1;0];
                sample(:,2)=[0;0];
        end
    case('1Dy')
        switch(lens.shape)
            case('circle')
                disp('warning this code does not work with 1D circle lenses')
```

```
case('rect')
    sample(:,1)=[0;0];
    sample(:,2)=sampling_scale*HPBW(2)*1.12*[0;1];
end

switch(lens.lensing)
case('2D')
    switch(lens.shape)
    case('circle')
        x_space=2*abs(sample(1,1));
        switch(fit)
        case('ceil')
            num_u1=ceil(theta_max/x_space);
            num_u2=num_u1;
            sample_new=theta_max/num_u1;
            sample=(sample_new/x_space)*sample;
        case('floor')
            num_u1=floor(theta_max/x_space);
            num_u2=num_u1;
        end
    case('rect')
        x_space=abs(sample(1,1));
        y_space=abs(sample(2,2));
        switch(fit)
        case('ceil')
            num_u1=ceil(theta_max/x_space);
            sample_new=theta_max/num_u1;
            sample(:,1)=(sample_new/x_space)*sample(:,1);
            num_u2=ceil(theta_max/y_space);
            sample_new=theta_max/num_u2;
            sample(:,2)=(sample_new/y_space)*sample(:,2);
        case('floor')
            num_u1=floor(theta_max/x_space);
            num_u2=floor(theta_max/y_space);
        end
    end

    pos=[];
    for k=-num_u1-1:num_u1+1
        for l=-num_u2-1:num_u2+1
            pos_temp=sample*[k;l];
            if sqrt(pos_temp'*pos_temp)/theta_max < 1.04
                pos=[pos,pos_temp];
            end
        end
    end

case('1Dx')
    switch(lens.shape)
    case('circle')
```

```
        disp('warning this code does not work with 1D circle lenses')
    case('rect')
        x_space=abs(sample(1,1));
        switch(fit)
        case('ceil')
            num_u1=ceil(theta_max/x_space);
            sample_new=theta_max/num_u1;
            sample(:,1)=(sample_new/x_space)*sample(:,1);
        case('floor')
            num_u1=floor(theta_max/x_space);
        end
    end

    pos=[];
    for k=-num_u1-1:num_u1+1
        pos_temp=sample*[k;0];
        if sqrt(pos_temp'*pos_temp)/theta_max < 1.04
            pos=[pos,pos_temp];
        end
    end

case('1Dy')
    switch(lens.shape)
    case('circle')
        disp('warning this code does not work with 1D circle lenses')
    case('rect')
        y_space=abs(sample(2,2));
        switch(fit)
        case('ceil')
            num_u2=ceil(theta_max/y_space);
            sample_new=theta_max/num_u2;
            sample(:,2)=(sample_new/y_space)*sample(:,2);
        case('floor')
            num_u2=floor(theta_max/y_space);
        end
    end

    pos=[];
    for k=-num_u2-1:num_u2+1
        pos_temp=sample*[0;k];
        if sqrt(pos_temp'*pos_temp)/theta_max < 1.04
            pos=[pos,pos_temp];
        end
    end
end

[m,n]=size(pos);

for k=1:n
    pos_temp=pos(:,k);
    if detector.N_c==1
```

```matlab
theta_temp=sqrt(pos_temp'*pos_temp);
if theta_temp>.01
    phi_temp=atan2(pos_temp(2),pos_temp(1));
    if phi_temp<0
        phi_temp=phi_temp+2*pi;
    end
else
    phi_temp=0;
end

focal_temp=get_focal(lens, theta_temp);
pos_xyz=[focal_temp*sin(theta_temp)*cos(phi_temp);...
         focal_temp*sin(theta_temp)*sin(phi_temp);...
         focal_temp*cos(theta_temp)];
a_z=(-1*pos_xyz')/sqrt(pos_xyz'*pos_xyz);
detector_temp=detector;
[antenna_temp]=build_ant(detector.cell.name{1},pos_xyz,...
          a_z,detector.cell.theta_r{1},detector.cell.polar{1});
detector_temp.cell.pos(1)=antenna_temp.pos(1);
detector_temp.cell.name(1)=antenna_temp.name(1);
detector_temp.cell.base(1)=antenna_temp.base(1);
detector_temp.cell.theta_r(1)=antenna_temp.theta_r(1);
detector_temp.cell.local(1)=antenna_temp.local(1);
detector_temp.cell.polar(1)=antenna_temp.polar(1);

imager{k}=detector_temp;
else
    disp('need to write code that places arrays')
end
end


_____ noiseTF.m _____

function [noise_TF]=noiseTF(lens,imager,optional)

%this function creates a transfer function that related the noise sources
%(amplifiers) to the detector outputs.
%noise_TF.lens is the lens sources
%noise_TF.det is cell structure containing the TF for each detectors unitcell
%noise_TF.imager is the noise source at each port of the imager before sampling
%The noise doesn't go through the amplifiers.
%optional.delay_lens ('before', 'after') default 'before'
%optional.delay_detector ('before', 'after') default 'before'
%optional.polar use polarization for feed coupling
%          default is 'yes'
%optional.pathloss use pathloss for feed coupling
%          default is 'yes'
%optional.flags flags the turn on/off the calculates of TF
%          [lens,detector,imager]
```

```matlab
%          default is [1,1,1]

delay_lens='before';
delay_detector='before';
optional_feed.polar='yes';
optional_feed.pathloss='yes';
flag=[1,1,1];

if nargin==3
    if isfield(optional,'delay_lens')
        delay_lens=optional.delay_lens;
    end

    if isfield(optional,'delay_detector')
        delay_detector=optional.delay_detector;
    end

    if isfield(optional,'polar')
        optional_feed.polar=optional.polar;
    end

    if isfield(optional,'pathloss')
        optional_feed.pathloss=optional.pathloss;
    end

    if isfield(optional,'flags')
        flags=optional.flags;
    end
end

if flags(1)==1
    for k=1:lens.N_c
        %comments(['QOL noise TF ',num2str(k/lens.N_c,3)]);
        for m=1:length(imager)
            detector=imager{m};
            [fd_antenna]=pull_antenna(lens,k,'feed');

            switch(delay_lens)
                case 'before'
                    lin_fd=exp(-j*2*pi*lens.cell.delay(k));
                case 'after'
                    lin_fd=1;
            end

            Voc=0;
            for l=1:detector.N_c
                [det_antenna]=pull_antenna(detector,l);
                [Voc_l]=couple(fd_antenna,det_antenna,lin_fd,optional_feed);
                switch(delay_detector)
                    case 'before'
                        Voc=Voc+Voc_l*detector.g*detector.cell.g{l}*...
```

```matlab
                exp(-j*2*pi*detector.cell.delay{l});
        case 'after'
            Voc=Voc+Voc_1*detector.g*detector.cell.g{l};
            end
        end
        Voc=Voc*detector.g;

        noise_TF.lens(m,k)=Voc;
    end
end

if flags(2)==1
    for k=1:length(imager)
        detector=imager{k};

        noise_det_temp=zeros(1,detector.N_c);

        for l=1:detector.N_c
            switch(delay,detector)
            case('before')
                noise_det_temp(1)=detector.g*exp(-j*2*pi*detector.cell.delay{l});
            case('after')
                noise_det_temp(1)=detector.g;
            end
        end
        noise_TF.det{k}=noise_det_temp;
    end
end

if flags(3)==1
    noise_TF.imager=eye(length(imager));
end
```

_____ number_elements.m _____

```matlab
function number_elements(device,optional)

%this function numbers the antennas in the device (lens detectors channel
%imager)
%optiona.fig_num is optional and default is 4
%optional.font_size this optional allows the fonts size to be changed
%           default is 10

fig_num=4;
font_size=10;
if nargin==2
    if isfield(optional,'fig_num')
```

```matlab
        fig_num=optional.fig_num;
    end
    if isfield(optional,'font_size')
        font_size=optional.font_size;
    end
end

figure(fig_num)

if findstr(class(device),'cell')
    for k=1:length(device)
        device_k=device{k};

        switch device_k.device
        case('detector')
            for m=1:device_k.N_c
                pos=device_k.cell.pos{m};
                [H_temp]=text(pos(1),-1*pos(3),pos(2),...
                    [num2str(k,2),',',num2str(m,3)]);
                set(H_temp,'FontSize',font_size)
            end
        case('source')
            pos=device_k.pos;
            switch device_k.type
            case('nearfield')
                [H_temp]=text(pos(1),-1*pos(3), pos(2),num2str(k,3));
                set(H_temp,'FontSize',font_size)
            case('farfield')
                [H_temp]=text(pos(1),-1*pos(3),pos(2),num2str(k,3));
                set(H_temp,'FontSize',font_size)
            end
        end
    end
else
    switch device.device
    case('lens')
        for m=1:device.N_c
            pos=device.cell.nfd.pos{m};
            [H_temp]=text(pos(1),-1*pos(3),pos(2),num2str(m,3));
            set(H_temp,'FontSize',font_size)
        end
    case('array')
        for m=1:device.N_c
            pos=device.cell.nfd.pos{m};
            [H_temp]=text(pos(1),-1*pos(3),pos(2),num2str(m,3));
            set(H_temp,'FontSize',font_size)
        end
    case('detector')
        for m=1:device.N_c
            pos=device.cell.pos{m};
            [H_temp]=text(pos(1),-1*pos(3),pos(2),num2str(m,3));
```

```
        set(H_temp,'FontSize',font_size);
    end
  end
end

_____ per_error.m _____

function [error_ptrn]=per_error(image_ptrn, mutual_ptrn)

%this function plots the percent error do to mutual coupling

error_ptrn=image_ptrn;
error_ptrn.create='per_error';
error_ptrn.error_point=abs(mutual_ptrn.Voc./image_ptrn.Voc);
temp_max=max(max(abs(image_ptrn.Voc)));
error_ptrn.error_max=abs(mutual_ptrn.Voc./(temp_max*ones(size(mutual_ptrn.Voc))));

_____ planar_array.m _____

function [array]=planar_array(shape,dimension,antenna,unitcellsize,latice)

%Shape is a string argument that describes the lens shape 'circle','rect'
%dimension is a vector that contains the important dimensions for the array
% circle [radius]
% rect [width,height] width = x dimension and height = y dimension
%antenna is cell describes the antenna types,
%polarization and orientation.  The polarization vector is for a model
%antenna in the oreintation for the theorical or measured radiation patterns.
%Orientation discribes the rotation of the antennas from the model antenna.
%unitcellsize defines the x/y dimensions for unitcell
% [width,height] width = x  height = y
%latice is a cell of strings that discribes the how the unitcells are
%placed in the lens {pattern,boundary}
% PATTERN
% 'triangle' uses a triangular lattice to fill the shape trying to maximise
% the number of unit cells
% 'max' uses a row structure with as many unitcells as possible in
% each row
% BOUNDARY
% 'best' is the tightest packing allowing for 1/4 of the unitcell
% to extend outside the defined shape
% 'limit' is the tightest packing without letting the unitcells
% extend outside the defined shape
%note: all dimensions are in wave length of the center frequency
%note: this code assumes for now that the lens is confined to a plane
```

```
%initialize array
array.device='array';
array.shape=shape;
array.dimension=dimension;
array.cell.width=unitcellsize(1);
array.cell.height=unitcellsize(2);

switch shape
  case {'circle'}
    [array]=packcircle(latice,array);
  case {'rect'}
    [array]=packrect(latice,array);
end

array.N_c=sum(array.R);
[array]=packant2(array,antenna);

_____ plot_antenna.m _____

function plot_antenna(antenna_ptrn,parameter,optional)

%this function plots the antenna_ptrn pwr or polarization
%parameter {'pattern','polar'}  the pattern is on a linear scale
%                               the polarization is plotted 1/8 of effective
%                               length circular polarization plots incorrectly
%optional.data {'raw', 'norm' ,'num' ,num}  default is the  normalize the max power
%                               to 0 dB or the max power can be normalize
%                               to 'num' in dB
%optional.export {'screen', 'file'}   this option ajusts the colors for
%                               screen viewing or file exporting

figure(7)
whitebg('K')
view(3)

data='norm';
export='screen';
if nargin==3
  if isfield(optional,'data')
    data=optional.data;
  end
  if isfield(optional,'export')
    export=optional.export;
  end
end

switch (parameter)
  case 'pattern'
    colormap('jet')
```

```
        end
      end
      hold off
      xlabel('x')
      ylabel('y')
      zlabel('z')
      axis('equal')

end

% _____ plot_array.m _____

function plot_array(array,optional)

%plot_array(array,optional)
%This function plots a 3d picture of the array in figure 4
%optional.fig_num default is 4

fig_num=4;
if nargin==2
    fig_num=optional.fig_num;
end

whitebg('k')
figure(fig_num)
view(3)

for m=1:array.N_c
    drawrect(array.cell.nfd.base(m),array.cell.nfd.pos(m),array.cell.height, ...
        array.cell.width,'r');

    hold on
    plot3(array.cell.nfd.pos(m)(1),-1*array.cell.nfd.pos(m)(3), ...
        array.cell.nfd.pos(m)(2),'rx');

end

hold on
switch array.shape
    case('circle')
        angle=[0:pi/15:2*pi];
        D=2*array.dimension;
        x_circle=(D/2)*cos(angle);
        y_circle=(D/2)*sin(angle);
        z_circle=zeros(1,length(angle));
        plot3(x_circle,-1*z_circle,y_circle,'r')
    case {'rect'}
        w=array.dimension(1);
        h=array.dimension(2);
        x=[-w/2, -w/2, w/2, w/2, -w/2];
```

```
pwr=antenna_ptrn.le.*antenna_ptrn.le;

max_pwr=max(max(pwr));

if strcmp(data,'norm')
    pwr=pwr/max_pwr;
elseif strcmp(class(data),'double')
    data=10^(data/10);
    pwr=(data/max_pwr)*pwr;
end

pwr=10*log10(pwr);

%

for k=1:length(antenna_ptrn.alpha)
    for l=1:length(antenna_ptrn.ki)
        z(k,l)=pwr(k,l)*cos(antenna_ptrn.alpha(k));
        y(k,l)=pwr(k,l)*sin(antenna_ptrn.alpha(k))*sin(antenna_ptrn.ki(l));
        x(k,l)=pwr(k,l)*sin(antenna_ptrn.alpha(k))*cos(antenna_ptrn.ki(l));
    end
end

surf(x,y,z,x.^2+y.^2+z.^2)
%colorbar
xlabel('x')
ylabel('y')
zlabel('z')
axis('equal')
case 'polar'
    drawball(1)
    colormap('gray');
    switch(export)
        case 'screen'
            caxis([1,1.1]);
        case 'file'
            caxis([0,0.1]);
    end
    hold on
    for k=1:length(antenna_ptrn.alpha)
        for l=1:length(antenna_ptrn.ki)
            if antenna_ptrn.alpha(k)==0 & antenna_ptrn.ki~=0

            else
                z_temp=cos(antenna_ptrn.alpha(k));
                y_temp=sin(antenna_ptrn.alpha(k))*sin(antenna_ptrn.ki(l));
                x_temp=sin(antenna_ptrn.alpha(k))*cos(antenna_ptrn.ki(l));
                [J_sph]=xyz2sph(antenna_ptrn.alpha(k),antenna_ptrn.ki(l));
                polar_vector= J_sph'*antenna_ptrn.polar(k,l)/8;
                plot3([x_temp,x_temp+polar_vector(1)],...
                    [y_temp,y_temp+polar_vector(2)],...
                    [z_temp,z_temp+polar_vector(3)],'g');
                plot3(x_temp,y_temp,z_temp,'go');
            end
```

```
y=[h/2, -h/2, -h/2, h/2, h/2];
z=zeros(1,5);
plot3(x,-1*z,y,'r')

end
hold off
axis('equal')

xlabel('x')
ylabel('-z')
zlabel('y')

_____ plot_channel.m _____

function plot_channel(channel,optional)

%This function plots a 3d picture of the channel in figure 4
%optional.fig_num default is 4
%optional.radius changes the size of source symbol radius
%   default is 0.25
%optional.distance changes the distance the far field source symbols
%   are plotted default is 6

optional_plot.fig_num=4;
optional_plot.radius=0.25;
if nargin==2
  if isfield(optional,'fig_num')
    optional_plot.fig_num=optional.fig_num;
  end
  if isfield(optional,'radius')
    optional_plot.radius=optional.radius;
  end
  if isfield(optional,'distance')
    optional_plot.source.pos=(optional.distance/6)*source.pos;
  end
end

for k=1:length(channel)
  source_k=channel{k};
  plot_source(source_k,optional_plot);
end

_____ plot_couple.m _____

function plot_couple(transmit,receive,optional)

%function plot_couple(transmit,receive,optional)
```

```
%this function plots the coupling between two antennas 'transmit' (red) and
%'receive' (blue). Only linear polarization and 1/r loss is used. All
%antenna fields are required (name,polar,bae,local,pos,theta_r).
%optional.pathloss default is 'yes'
%optional.fig_num default is 5

fig_num=5;
pathloss='yes';
if nargin==3
  if isfield(optional,'pathloss')
    pathloss=optional.pathloss;
  end

  if isfield(optional,'fig_num')
    fig_num=optional.fig_num;
  end

end

figure(fig_num)
view(3)
whitebg('k')

drawrect(transmit.local,transmit.pos,.5,.5,'r')
drawrect(receive.local,receive.pos,.5,.5,'g')

drawunit(transmit.local,transmit.pos,'r')
drawunit(receive.local,receive.pos,'g')

[theta_rec,phi_rec]=get_direction(receive,transmit);
[theta_tran,phi_tran]=get_direction(transmit,receive);
J_sph_tran=xyz2sph(theta_tran,phi_tran);
J_sph_rec=xyz2sph(theta_rec,phi_rec);
J_gt=transmit.local;
J_gr=receive.local;

Rt=transmit.pos;
Rr=receive.pos;
r=Rr-Rt;
r_mag=sqrt(r'*r);

[le_t,ro_t]=feval(transmit.name,theta_tran,phi_tran,transmit.polar);
ro_t_l=J_gt'*J_sph_tran'*ro_t;

switch pathloss
  case ('no')
    Ei_g=((2*pi)/(4*pi))*le_t*ro_t_l;
  case('yes')
    Ei_g=((2*pi)/(4*pi*r_mag))*le_t*ro_t_l;
end

pos=receive.pos;
```

```
function plot_detector_pos(imager)

%this function plots detectors positions on image plot

figure(2);
whitebg('w')
for k=1:length(imager)
    detector_k=imager{k};
    pos_k=detector_k.cell.pos{1};

    theta_k=atan2(sqrt(pos_K(1)^2+pos_K(2)^2),pos_K(3));

    if theta_k > .001
        phi_k=atan2(pos_k(2),pos_k(1));
        if phi_k < 0
            phi_k=phi_k+2*pi;
        end
    else
        phi_k=0;
    end

    x_k=theta_k*cos(phi_k);
    y_k=theta_k*sin(phi_k);
    hold on
    plot3(x_k,y_k,0,'kx')
    text(x_k,y_k,0,num2str(k,3))
    hold off

end
```

_____ plot_error.m _____

```
function [error_ptrn_plot]=plot_error(error_ptrn,ref,optional)

%this function plots the error pattern and outputs the plotted data
%'ref' determines which percent error calculation to plot
%    'point' is point by point calculation
%    'max' percent error referenced to max power of image pattern
%optional.scale ('linear','dB') default is dB
%optional.data ('raw',num) default  num is the max scale limit
%optional.format ('polar','rect') default is polar representation of
%                  theta and phi where the imageius is
%                  linear with theta
%optional.theta_range [theta_start,theta_end] if theta_start = theta_end
%                     it plots is a cut with theta=theta_start
%optional.phi_range [phi_start,phi_end] if phi_start = phi_end then it produces
%                     the plot is a cut with phi=phi_start
%if either of the theta or phi ranges are of length one, then plot_image
```

```
xpoint=(E1_g)+pos;

hold on
plot3([pos(1),xpoint(1)],-1*[pos(3),xpoint(3)],[pos(2),xpoint(2)],'y')
plot3([transmit.pos(1),receive.pos(1)],-1*[transmit.pos(3),receive.pos(3)], ...
    [transmit.pos(2),receive.pos(2)],'b')

axis('equal')

hold off
grid on
```

_____ plot_detector.m _____

```
function plot_detector(detector,optional)

%this function plots a 3d picture of the detector in figure 4
%optional.fig_num default is 4

fig_num=4;
if nargin==2
    fig_num=optional.fig_num;
end

whitebg('k')
figure(fig_num)
view(3)
connecting=[];
for m=1:detector.N_c
    drawrect(detector.cell.base{m},detector.cell.pos{m},detector.cell.height, ...
        detector.cell.width,'g');
    hold on
    plot3(detector.cell.pos{m}(1),-1*detector.cell.pos{m}(3), ...
        detector.cell.pos{m}(2),'rx');
    hold off
    connecting=[connecting,detector.cell.pos{m}];
end
hold on
plot3(connecting(1,:),-1*connecting(3,:),connecting(2,:),'g')
hold off
axis('equal')
```

_____ plot_detector_pos.m _____

```matlab
%allows stacking of 2D plots and optional.line_color defines the line
%type and color (default is blue solid)
%if the output of one plot_image is feed into another plot_image
%then the data is reploted ignoring all optional's
%if measured data if feed is used, then the optional.freq determines
%which pattern is used

scale='linear';
data=100;
format='polar';
figure(5)
line_color='b-';

if strcmp(ref,'point')
    error_ptrn.error=error_ptrn.error_point;
elseif strcmp(ref,'max')
    error_ptrn.error=error_ptrn.error_max;
end

if nargin==3
    if isfield(optional,'line_color')
        line_color=optional.line_color;
    end
end

switch (error_ptrn.create)
    case('plot_error')
        if isfield(error_ptrn,'x')
            hold on
            plot(error_ptrn.x,error_ptrn.y,error_ptrn.error)
            surf(error_ptrn.x,error_ptrn.y,error_ptrn.error,line_color)
            title(['theta is from ',num2str(error_ptrn.theta_range(1)/pi,2),'-'...
                num2str(error_ptrn.theta_range(2)/pi,2),' (pi)'])
            xlabel('x')
            ylabel('y')
            zlabel(['percent error (',scale,')'])
            axis('square')
        else
            if length(error_ptrn.theta)==1
                hold on
                plot(error_ptrn.phi/pi,error_ptrn.error,line_color)
                title(['theta = ',num2str(error_ptrn.theta(1))]);
                xlabel('phi (pi)')
                ylabel(['pwr (',scale,')'])
                hold off
            elseif length(error_ptrn.phi)==1
                hold on
                plot(error_ptrn.theta/pi,error_ptrn.error,line_color)
                title(['theta = ',num2str(error_ptrn.phi(1))]);
                xlabel('theta (pi)')
                ylabel(['percent error (',scale,')'])
                hold off
            else
```

```matlab
                surf(error_ptrn.phi/pi,error_ptrn.theta/pi,error_ptrn.error);
                xlabel('phi (pi)')
                ylabel('theta (pi)')
                zlabel(['percent error (',scale,')'])
            end
    case('per_error')
        if nargin ==3
            if isfield(optional,'scale')
                scale=optional.scale;
            end
            if isfield(optional,'data')
                data=optional.data;
            end
            if isfield(optional,'format')
                format=optional.format;
            end
            if isfield(optional,'theta_range')
                theta_range=optional.theta_range;
                [Y,start_temp]=min(abs(error_ptrn.theta-theta_range(1)*...
                    ones(size(error_ptrn.theta))));
                [Y,end_temp]=min(abs(error_ptrn.theta-theta_range(2)*...
                    ones(size(error_ptrn.theta))));
                error_ptrn.theta=error_ptrn.theta([start_temp:end_temp]);
                error_ptrn.error=error_ptrn.error([start_temp:end_temp],:);
            end
            if isfield(optional,'phi_range')
                phi_range=optional.phi_range;
                [Y,start_temp]=min(abs(error_ptrn.phi-phi_range(1)*...
                    ones(size(error_ptrn.phi))));
                [Y,end_temp]=min(abs(error_ptrn.phi-phi_range(2)*...
                    ones(size(error_ptrn.phi))));
                error_ptrn.phi=error_ptrn.phi([start_temp:end_temp]);
                error_ptrn.error=error_ptrn.error(:,[start_temp:end_temp]);
            end
        end
        error_data=error_ptrn.error;
        if strcmp(class(data),'double')
            [m,n]=size(error_data);
            for k=1:m
                for l=1:n
                    if error_data(k,l)>data
                        error_data(k,l)=data;
                    end
                end
            end
        end
        if strcmp(scale,'dB')
```

```matlab
      error_ptrn_plot.theta=error_ptrn.theta;
      error_ptrn_plot.phi=error_ptrn.phi;
    end
  end
  error_ptrn_plot.create'plot_error';
end
```

```
_____ plot_focal.m _____
```

```matlab
function plotfocl(lens,optional)

%plotfocl(lens,max_theta)
%This function plots the focal surface
%optional.surface ['focal','radius] chooses between the focal surface and
%                   a sphere of radius of radius 'radius'
%                   default is 'focal'
%optional.theta_max theta_max is the max angle for ploting the focal surface
%                   default is theta_max=pi/4
%optional.rings   the number of ring plotted
%                   default is 6
%optional.fig_num is the numb of the figure the surface will be plotted in
%                   default is figure 4
%optional.color is the color for the surface
%                   default is 'c' cyan

surface='focal';
theta_max=pi/4;
rings=6;
fig_num=4;
color='c';

if nargin==2
  if isfield(optional,'surface')
    surface=optional.surface;
  end
  if isfield(optional,'theta_max')
    theta_max=optional.theta_max;
  end
  if isfield(optional,'rings')
    rings=optional.rings;
  end
  if isfield(optional,'fig_num')
    fig_num=optional.fig_num;
  end
  if isfield(optional,'color')
    color=optional.color;
  end
end
```

```matlab
    error_data=20*log10(error_data);
end

if length(error_ptrn.theta)==1
  hold on
  plot(error_ptrn.phi/pi,pwr,line_color)
  title(['theta = ',num2str(error_ptrn.theta(1))]);
  xlabel('phi (pi)')
  ylabel(['percent error (',scale,')'])
  error_ptrn_plot.error=error_data;
  error_ptrn_plot.theta=error_ptrn.theta;
  error_ptrn_plot.phi=error_ptrn.phi;
  hold off
elseif length(error_ptrn.phi)==1
  hold on
  plot(error_ptrn.theta/pi,pwr,line_color)
  title(['theta = ',num2str(error_ptrn.phi(1))]);
  xlabel('theta (pi)')
  ylabel(['percent error (',scale,')'])
  error_ptrn_plot.error=error_data;
  error_ptrn_plot.theta=error_ptrn.theta;
  error_ptrn_plot.phi=error_ptrn.phi;
  hold off
else
  switch (format)
    case ('polar')
      for k=1:length(error_ptrn.theta)
        for l=1:length(error_ptrn.phi)
          x(k,l)=error_ptrn.theta(k)*cos(error_ptrn.phi(1));
          y(k,l)=error_ptrn.theta(k)*sin(error_ptrn.phi(1));
        end
      end
      surf(x,y,error_data)
      title(['theta is from ',num2str(error_ptrn.theta(1)/pi,2),'-'...
        num2str(error_ptrn.theta(length(error_ptrn.theta))/pi,2),...
        ' (pi)'])
      xlabel('x')
      ylabel('y')
      zlabel(['percent error(',scale,')'])
      axis('square')
      error_ptrn_plot.error=error_data;
      error_ptrn_plot.x=x;
      error_ptrn_plot.y=y;
      error_ptrn_plot.theta_range=[error_ptrn.theta(1),...
        error_ptrn.theta(length(error_ptrn.theta))];
    case('rect')
      surf(error_ptrn.phi/pi,error_ptrn.theta/pi,pwr);
      xlabel('phi (pi)')
      ylabel('theta (pi)')
      zlabel(['percent error (',scale,')'])
      error_ptrn_plot.error=error_data;
```

```
%optional.scale  ('linear','dB') default is dB
%optional.data  ('raw','norm',num) default is normalize to max unity
%                                   num is the normalizaton of the max power
%                                   to this value in the scan range and
%                                   scale specified
%optional.format ('polar','rect') default is polar representation of
%                                   theta and phi where the imageius is
%                                   linear with theta
%optional.theta_range [theta_start,theta_end] if theta_start = theta_end
%                                   it plots is a cut with theta=theta_start
%optional.phi_range [phi_start,phi_end] if phi_start = phi_end then it produces
%                                   the plot is a cut with phi=phi_start
%if either of the theta or phi ranges are of length one, then plot_image
%allows stacking of 2D plots and optional.line_color defines the line
%type and color (default is blue solid)
%if the output of one plot_image is feed into another plot_image
%then the data is reploted ignoring all optional's
%if measured data if feed is used, then the optional.freq determines
%which pattern is used

scale='dB';
data='norm';
format='polar';
figure(2)
line_color='b-';

if nargin==2
    if isfield(optional,'line_color')
        line_color=optional.line_color;
    end
end

switch (image_ptrn.create)
    case('plot_image')
        if isfield(image_ptrn,'x')
            surf(image_ptrn.x,image_ptrn.y,image_ptrn.pwr)
            title(['theta is from ',num2str(image_ptrn.theta_range(1)/pi,2),'-',...
                num2str(image_ptrn.theta_range(2)/pi,2),' (pi)'])
            xlabel('x')
            ylabel('y')
            zlabel(['pwr (',scale,')'])
            axis('square')
        else
            if length(image_ptrn.theta)==1
                hold on
                plot(image_ptrn.phi/pi,image_ptrn.pwr,line_color)
                title(['theta = ',num2str(image_ptrn.theta(1))]);
                xlabel('phi (pi)')
                ylabel(['pwr (',scale,')'])
                hold off
            elseif length(image_ptrn.phi)==1
```

```
figure(fig_num)
view(3)
phi=[0:pi/10:2*pi];
theta=[theta_max/rings:theta_max/rings:theta_max];
phi_line=[0:pi/4:2*pi];

X_line=[];
Y_line=[];
Z_line=[];
for k=1:length(theta)
    if strcmp(surface,'focal')
        focal_rad=get_focal(lens,theta(k));
    else
        focal_rad=surface;
    end

    x_circle=-1*focal_rad*sin(theta(k))*cos(phi);
    y_circle=-1*focal_rad*sin(theta(k))*sin(phi);
    z_circle=focal_rad*cos(theta(k))*ones(1,length(phi));

    hold on
    plot3(x_circle,-1*z_circle,y_circle,color)
    hold off

    x_temp=-1*focal_rad*sin(theta(k))*cos(phi_line);
    y_temp=-1*focal_rad*sin(theta(k))*sin(phi_line);
    z_temp=focal_rad*cos(theta(k))*ones(1,length(phi_line));

    X_line=[X_line;x_temp];
    Y_line=[Y_line;y_temp];
    Z_line=[Z_line;z_temp];

end

hold off
xlabel('x')
ylabel('-z')
zlabel('y')
axis('equal')

hold on

    plot3(X_line,-1*Z_line,Y_line,color)

hold off

_____ plot_image.m _____

function [image_ptrn_plot]=plot_image(image_ptrn,optional)

%this function plots the image pattern and outputs the plotted data
```

```
    Hold on
    plot(image_ptrn.theta/p1,image_ptrn.pwr,line_color)
    title(['theta = ',num2str(image_ptrn.phi(1))]);
    xlabel('theta (p1)')
    ylabel(['pwr (',scale,')'])
    hold off
  else
    surf(image_ptrn.phi/p1,image_ptrn.theta/p1,image_ptrn.pwr);
    xlabel('phi (p1)')
    ylabel('theta (p1)')
    zlabel(['pwr (',scale,')'])
  end

case('image_pattern','measured')
  if strcmp(image_ptrn.create,'measured')
    %pull image pattern for optional.freq
    %squareroot the pwr and make field image_ptrn.Voc
    disp('measure data part of the code is not done yet')
  end

if nargin ==2
  if isfield(optional,'scale')
    scale=optional.scale;
  end
  if isfield(optional,'data')
    data=optional.data;
  end
  if isfield(optional,'format')
    format=optional.format;
  end
  if isfield(optional,'theta_range')
    theta_range=optional.theta_range;
    [Y,start_temp]=min(abs(image_ptrn.theta-theta_range(1)*...
      ones(size(image_ptrn.theta))));
    [Y,end_temp]=min(abs(image_ptrn.theta-theta_range(2)*...
      ones(size(image_ptrn.theta))));
    image_ptrn.theta=image_ptrn.theta([start_temp:end_temp]);
    image_ptrn.Voc=image_ptrn.Voc([start_temp:end_temp],:);
  end
  if isfield(optional,'phi_range')
    phi_range=optional.phi_range;
    [Y,start_temp]=min(abs(image_ptrn.phi-phi_range(1)*...
      ones(size(image_ptrn.phi))));
    [Y,end_temp]=min(abs(image_ptrn.phi-phi_range(2)*...
      ones(size(image_ptrn.phi))));
    image_ptrn.phi=image_ptrn.phi([start_temp:end_temp]);
    image_ptrn.Voc=image_ptrn.Voc(:,[start_temp:end_temp]);
  end
end

pwr=(abs(image_ptrn.Voc)).^2;
```

```
  max_value=max(max(pwr));
  max_temp=max_value;

  if strcmp(data,'norm')
    max_temp=1;
  elseif strcmp(class(data),'double')
    if strcmp(scale,'linear')
      max_temp=data;
    else
      max_temp=10^(data/10);
    end
  end

  pwr=max_temp*pwr/max_value;

  if strcmp(scale,'dB')
    pwr=10*log10(pwr);
  end

  if length(image_ptrn.theta)==1
    hold on
    plot(image_ptrn.phi/p1,pwr,line_color)
    title(['theta = ',num2str(image_ptrn.theta(1))]);
    xlabel('phi (p1)')
    ylabel(['pwr (',scale,')'])
    image_ptrn_plot.pwr=pwr;
    image_ptrn_plot.theta=image_ptrn.theta;
    image_ptrn_plot.phi=image_ptrn.phi;
    hold off
  elseif length(image_ptrn.phi)==1
    hold on
    plot(image_ptrn.theta/p1,pwr,line_color)
    title(['theta = ',num2str(image_ptrn.phi(1))]);
    xlabel('theta (p1)')
    ylabel(['pwr (',scale,')'])
    image_ptrn_plot.pwr=pwr;
    image_ptrn_plot.theta=image_ptrn.theta;
    image_ptrn_plot.phi=image_ptrn.phi;
    hold off
  else
    switch (format)
      case ('polar')
        for k=1:length(image_ptrn.theta)
          for l=1:length(image_ptrn.phi)
            x(K,1)=image_ptrn.theta(K)*cos(image_ptrn.phi(1));
            y(K,1)=image_ptrn.theta(K)*sin(image_ptrn.phi(1));
          end
        end
        surf(x,y,pwr)
        title(['theta is from ',num2str(image_ptrn.theta(1)/p1,2),'-',...
```

```
        num2str(image_ptrn.theta(length(image_ptrn.theta))/pi,2),...
        , (pi)'])
    xlabel('x')
    ylabel('y')
    zlabel(['pwr (',scale,')'])
    axis('square')
    image_ptrn_plot.pwr=pwr;
    image_ptrn_plot.x=x;
    image_ptrn_plot.y=y;
    image_ptrn_plot.theta_range=[image_ptrn.theta(1),...
        image_ptrn.theta(length(image_ptrn.theta))];
    case('rect')
    surf(image_ptrn.phi/pi,image_ptrn.theta/pi,pwr);
    xlabel('phi (pi)')
    ylabel('theta (pi)')
    zlabel(['pwr (',scale,')'])
    image_ptrn_plot.pwr=pwr;
    image_ptrn_plot.theta=image_ptrn.theta;
    image_ptrn_plot.phi=image_ptrn.phi;
    end
end
image_ptrn_plot.create='plot_image';
end
```

--------------------------- plot_imager.m ---------------------------

```
function plot_imager(imager,optional)

%This function plots a 3d picture of the imager in figure 4
%optional.fig_num default is 4

optional_plot.fig_num=4;
if nargin==2
    optional_plot.fig_num=optional.fig_num;
end

for k=1:length(imager)
    detector_k=imager{k};
    plot_detector(detector_k,optional_plot);
end
```

--------------------------- plot_lens.m ---------------------------

```
function plotlens(lens,optional)

%plotlens(lens,optional)
%This function plots a 3d picture of the quasi-optical lens in figure 4
```

```
%optional.fig_num default is 4

fig_num=4;
if nargin==2
    fig_num=optional.fig_num;
end

whitebg('k')
figure(fig_num)
view(3)

for m=1:lens.N_c
    drawrect(lens.cell.nfd.base{m},lens.cell.nfd.pos{m},lens.cell.cell.height, ...
        lens.cell.width,'r');

    hold on
    plot3(lens.cell.nfd.pos{m}(1),-1*lens.cell.nfd.pos{m}(3), ...
        lens.cell.nfd.pos{m}(2),'rx');

    plot3(lens.cell.fd.pos{m}(1),-1*lens.cell.fd.pos{m}(3), ...
        lens.cell.fd.pos{m}(2),'yo');
    hold off

end

hold on
switch lens.shape
    case('circle')
        angle=[0:pi/15:2*pi];
        D=2*lens.dimension;
        x_circle=(D/2)*cos(angle);
        y_circle=(D/2)*sin(angle);
        z_circle=zeros(1,length(angle));
        plot3(x_circle,-1*z_circle,y_circle,'r')
    case {'rect'}
        w=lens.dimension(1);
        h=lens.dimension(2);
        x=[-w/2, -w/2, w/2, w/2, -w/2];
        y=[h/2, -h/2, -h/2, h/2, h/2];
        z=zeros(1,5);
        plot3(x,-1*z,y,'r')

end
hold off
axis('equal')

xlabel('x')
ylabel('-z')
zlabel('y')
```

```matlab
function [rad_ptrn_plot]=plot_radiation(rad_ptrn,optional)

%this function plots the radiation pattern and outputs the ploted data
%optional.scale  ('linear','dB') default is dB
%optional.data ('raw','norm',num) default is normalize to max unity
%                    num is the normalizaton of the max power
%                    to this value in the scan range and
%                    scale specified
%optional.format ('polar','rect') default is polar representation of
%                    alpha and Ki where the radius is
%                    linear with alpha
%optional.alpha_range [alpha_start,alpha_end] if alpha_start = alpha_end
%                    it plots is a cut with alpha=alpha_start
%optional.Ki_range [Ki_start,Ki_end] if Ki_start = Ki_end then it produces
%                    the plot is a cut with Ki=Ki_start
%if either of the alpha or Ki ranges are of length one, then plot-radiati
%allows stacking of 2D plots and optional.line_color defines the line
%tye and color (default is blue solid)
%if the output of one plot_radiation is feed into another plot_radiation
%then the data is reploted ignoring all optional's
%if measured data if feed is used, then the optional.freq determines
%which pattern is used

scale='dB';
data='norm';
format='polar';
figure(3)
line_color='b-';

switch (rad_ptrn.create)
  case('plot_radiation')
    if isfield(rad_ptrn,'x')
      surf(rad_ptrn.x,rad_ptrn.y,rad_ptrn.pwr)
      title(['alpha is from ',num2str(rad_ptrn.alpha_range(1)/pi,2),'-',...
          num2str(rad_ptrn.alpha_range(2)/pi,2),' (pi)'])
      xlabel('x')
      ylabel('y')
      zlabel(['power (',scale,')'])
      axis('square')
    else
      if length(rad_ptrn.alpha)==1
        hold on
        plot(rad_ptrn.Ki/pi,rad_ptrn.pwr,line_color)
        title(['alpha = ',num2str(rad_ptrn.alpha(1))]);
        xlabel('Ki (pi)')
        ylabel(['power (',scale,')'])
        hold off
      elseif length(rad_ptrn.Ki)==1
```

```matlab
        hold on
        plot(rad_ptrn.alpha/pi,rad_ptrn.pwr,line_color)
        title(['alpha = ',num2str(rad_ptrn.Ki(1))]);
        xlabel('alpha (pi)')
        ylabel(['power (',scale,')'])
        hold off
      else
        surf(rad_ptrn.Ki/pi,rad_ptrn.alpha/pi,rad_ptrn.pwr);
        xlabel('Ki (pi)')
        ylabel('alpha (pi)')
        zlabel(['power (',scale,')'])
      end
  case('rad_pattern','measured','antenna_pattern')
    switch rad_ptrn.create
      case('measured')
        %pull radiation pattern for optional.freq
        %squareroot the pwr and make field rad_ptrn.Voc
        disp('measure data part of the code is not done yet')
      case('antenna_pattern')
        rad_ptrn.Voc=rad_ptrn.le;
    end

    if nargin==2
      if isfield(optional,'scale')
        scale=optional.scale;
      end
      if isfield(optional,'line_color')
        line_color=optional.line_color;
      end
      if isfield(optional,'data')
        data=optional.data;
      end
      if isfield(optional,'format')
        format=optional.format;
      end
      if isfield(optional,'alpha_range')
        alpha_range=optional.alpha_range
        [Y,start_temp]=min(abs(rad_ptrn.alpha-alpha_range(1)*...
            ones(size(rad_ptrn.alpha))));
        [Y,end_temp]=min(abs(rad_ptrn.alpha-alpha_range(2)*...
            ones(size(rad_ptrn.alpha))));
        rad_ptrn.alpha=rad_ptrn.alpha([start_temp:end_temp]);
        rad_ptrn.Voc=rad_ptrn.Voc([start_temp:end_temp],:);
      end
      if isfield(optional,'Ki_range')
        Ki_range=optional.Ki_range;
        if Ki_range(1)==Ki_range(2)
          Ki_range(2)=Ki_range(2)+pi;
          [Y,start_temp]=min(abs(rad_ptrn.Ki-Ki_range(1)*...
              ones(size(rad_ptrn.Ki))));
      end
```

```
    [Y,end_templ]=min(abs(rad_ptrn.Ki-Ki_range(2)*...
        ones(size(rad_ptrn.Ki))));
    rad_ptrn.Ki=Ki-Ki_range(1);
    rad_ptrn_start=rad_ptrn.Voc(:,[start_templ]);
    rad_ptrn_end=rad_ptrn.Voc(:,[end_templ]);
    rad_ptrn.Voc=[rad_ptrn_end([length(rad_ptrn.alpha):-1:1]);...
        rad_ptrn_start];
    rad_ptrn.alpha=...
        [-1*rad_ptrn.alpha([length(rad_ptrn.alpha):-1:1])...
        ,rad_ptrn.alpha];
    else

    [Y,start_templ]=min(abs(rad_ptrn.Ki-Ki_range(1)*...
        ones(size(rad_ptrn.Ki))));
    [Y,end_templ]=min(abs(rad_ptrn.Ki-Ki_range(2)*...
        ones(size(rad_ptrn.Ki))));
    rad_ptrn.Ki=rad_ptrn.Ki([start_templ:end_templ]);
    rad_ptrn.Voc=rad_ptrn.Voc(:,[start_templ:end_templ]);
    end
end

pwr=(abs(rad_ptrn.Voc)).^2;

max_value=max(max(pwr));
max_temp=max_value;

if strcmp(data,'norm')
    max_temp=1;
elseif strcmp(class(data),'double')
    if strcmp(scale,'linear')
        max_temp=data;
    else
        max_temp=10^(data/10);
    end
end

pwr=max_temp*pwr/max_value;

if strcmp(scale,'dB')
    pwr=10*log10(pwr);
end

if length(rad_ptrn.alpha)==1
    hold on
    plot(rad_ptrn.Ki/pi,pwr,line_color)
    title(['alpha = ',num2str(rad_ptrn.alpha(1))]);
    xlabel('chi (pi)')
    ylabel(['power (',scale,')'])
    rad_ptrn_plot.pwr=pwr;
```

```
    rad_ptrn_plot.alpha=rad_ptrn.alpha;
    rad_ptrn_plot.Ki=rad_ptrn.Ki;
    hold off
elseif length(rad_ptrn.Ki)==1
    hold on
    plot(rad_ptrn.alpha*180/pi,pwr,line_color)
    title(['alpha = ',num2str(rad_ptrn.Ki(1))]);
    xlabel('alpha (degrees)')
    ylabel(['power (',scale,')'])
    rad_ptrn_plot.pwr=pwr;
    rad_ptrn_plot.alpha=rad_ptrn.alpha;
    rad_ptrn_plot.Ki=rad_ptrn.Ki;
    hold off
else
    switch (format)
    case ('polar')
        for k=1:length(rad_ptrn.alpha)
            for l=1:length(rad_ptrn.Ki)
                x(k,l)=rad_ptrn.alpha(k)*cos(rad_ptrn.Ki(1));
                y(k,l)=rad_ptrn.alpha(k)*sin(rad_ptrn.Ki(1));
            end
        end
        surf(x,y,pwr)
        title(['alpha is from ',num2str(rad_ptrn.alpha(1)/pi,2),'-',...
            num2str(rad_ptrn.alpha(length(rad_ptrn.alpha))/pi,2),...
            ', (pi)'])
        xlabel('x')
        ylabel('y')
        zlabel(['power (',scale,')'])
        axis('square')
        rad_ptrn_plot.pwr=pwr;
        rad_ptrn_plot.x=x;
        rad_ptrn_plot.y=y;
        rad_ptrn_plot.alpha_range=[rad_ptrn.alpha(1),...
            rad_ptrn.alpha(length(rad_ptrn.alpha))];
    case ('rect')
        surf(rad_ptrn.Ki/pi,rad_ptrn.alpha/pi,pwr);
        xlabel('chi (pi)')
        ylabel('alpha (pi)')
        zlabel(['power (',scale,')'])
        rad_ptrn_plot.pwr=pwr;
        rad_ptrn_plot.alpha=rad_ptrn.alpha;
        rad_ptrn_plot.Ki=rad_ptrn.Ki;
    end

end
rad_ptrn_plot.create='plot_radiation';

end
```

```
function plot_source(source,optional)

%this function plots a 3d picture of the source in figure 4
%optional.fig_num default is 4
%optional.radius changes the size of source symbol radius
%    default is 0.25
%optional.distance changes the distance the far field source symbols
%    are plotted default is 6

fig_num=4;
radius=0.25;
if nargin==2
  if isfield(optional,'fig_num')
    fig_num=optional.fig_num;
  end
  if isfield(optional,'radius')
    radius=optional.radius;
  end
  if isfield(optional,'distance')
    source.pos=(optional.distance/6)*source.pos;
  end
end

whitebg('k')
figure(fig_num)
view(3)

switch(source.type)
  case('nearfield')
    drawpolygon(source.base,source.pos,radius,20,'m');
  case('farfield')
    drawpolygon(source.base,source.pos,radius,6,'m');
end

hold on
plot3(source.pos(1),-1*source.pos(3),source.pos(2),'rx');
axis('equal')
hold off
grid on
```

_____ plot_source_pos.m _____

```
function plot_source_pos(channel,name,offset)

%this function plots the signals on the radiation plattern plots
%the numbering is signals,source farfield 'x' and nearfield 'o'
%'name is the name of the source

figure(3)
```

```
whitebg('w')
for l=1:length(channel)
  source_l=channel(l);
  name=num2str(l,3);
  hold on
  switch (source_l.type)
    case('farfield')
      angle_l=source_l.angle;
      x=angle_l(1)*cos(angle_l(2));
      y=angle_l(1)*sin(angle_l(2));
      Htemp=plot3(x,y,0,'kx');
      set(Htemp,'MarkerSize',10);
      if nargin>1
        text(x+offset(1),y+offset(2),0,name)
      else
        text(x,y,0,[num2str(l,3)]);
      end
    case('nearfield')
      pos_l=source_l.pos
      alpha=atan2(sqrt(pos_l(1)^2+pos_l(2)^2),pos_l(3));
      Ki=atan2(pos_l(2),pos_l(1))
      if Ki<0
        Ki=Ki+2*pi;
      end
      x=angle_l(alpha)*cos(angle_l(Ki));
      y=angle_l(alpha)*sin(angle_l(Ki));
      Htemp=plot3(x,y,0,'ko');
      set(Htemp,'MarkerSize',10);
      if nargin>1
        text(x+offset(1),y+offset(2),0,name)
      else
        text(x,y,0,[num2str(l,3)]);
      end
  end
hold off
end
```

_____ ptrn_grid.m _____

```
function ptrn_grid(fig_num,angle_max,spacing,labels)

figure(fig_num);
angles=[spacing:spacing:angle_max];
theta=[0:pi/20:2*pi];

for k=1:length(angles)
  hold on
  for l=1:length(theta)
    x(l)=angles(k)*cos(theta(l));
```

```matlab
        y(1)=angles(k)*sin(theta(1));
        z(1)=1;
      end
      plot3(x,y,z,'K:')
      if strcmp(labels,'x')
        Htext=text(angles(k),0.1,1,[num2str(angles(k)*180/pi ,2),'~o']);
      else
        Htext=text(0.0,angles(k),1,[num2str(angles(k)*180/pi ,2),'~o']);
      end
      set(Htext,'Color',[0,0,0]);
      hold off
    end
    hold on
    plot3([0,angle_max],[0,0],[1,1],'b:')
    plot3([0,0],[0,angle_max],[1,1],'b:')
    plot3([0,-angle_max],[0,0],[1,1],'b:')
    plot3([0,0],[0,-angle_max],[1,1],'b:')
    hold off
```

_____ rad_pattern.m _____

```matlab
function [rad_ptrn]=rad_pattern(lens,detector,alpha_range,Ki_range,optional)

%this function outputs the receive Voc at a detector
%for a source with unit input.  'rad_ptrn' contains fields 'Voc', 'alpha'
%and 'Ki'.
%alpha_range is a vector that contains[alpha_start,alpha_end,alpha_N]
%Ki_range is a vector that contains[Ki_start,Ki_end,Ki_N]
%optional.polar_source use polarization for source coupling
%                      default is 'yes'
%optional.polar_feed use polarization for feed coupling
%                      default is 'yes'
%optional.pathloss_source use pathloss for source coupling
%                      default is 'yes'
%optional.pathloss_feed use pathloss for feed coupling
%                      default is 'yes'

optional_focus.polar_source='yes';
optional_focus.polar_feed='yes';
optional_focus.pathloss_source='yes';
optional_focus.pathloss_feed='yes';

%initalize default farfield source
middle=round(lens.N_c/2);
[nfd_antenna]=pull_antenna(lens,middle,'nonfeed');
[le_nfd,ro_nfd]=feval(nfd_antenna.name,0,0,nfd_antenna.polar);
J_sph_nfd=xyz2sph(0,0);
J_gnfd=nfd_antenna.local;
ro_source=J_gnfd'*J_sph_nfd*ro_nfd;
```

```matlab
polar=xyz2sph(0,0)*ro_source;
%

if nargin==5
  if isfield(optional,'polar_source')
    optional_focus.polar_source=optional.polar_source;
  end
  if isfield(optional,'polar_feed')
    optional_focus.polar_feed=optional.polar_feed;
  end
  if isfield(optional,'pathloss_source')
    optional_focus.pathloss_source=optional.pathloss_source;
  end
  if isfield(optional,'pathloss_feed')
    optional_focus.pathloss_feed=optional.pathloss_feed;
  end
end

if Ki_range(3)==1
  Ki=(1);
  alpha=[-alpha_range(2):(2*alpha_range(2))/(alpha_range(3)-1):...
    alpha_range(2)];
else
  alpha=[alpha_range(1):(alpha_range(2)-alpha_range(1))/(alpha_range(3)-1):...
    alpha_range(2)];
  Ki=[Ki_range(1):(Ki_range(2)-Ki_range(1))/(Ki_range(3)-1):Ki_range(2)];
end

for k=1:alpha_range(3)
  comments(['Radiation Pattern ',num2str(k/alpha_range(3),2)]);
  for l=1:Ki_range(3)
    if alpha(k)==0 & Ki(1)~=0
      Voc(k,1)=Voc(1,1);
    else
      [source_ki]=build_far(alpha(k),Ki(1),0,1,polar);
      [Voc(k,1)]=lens_focus(lens,source_ki,detector,optional_focus);
    end
  end
end

rad_ptrn.Voc=Voc;
rad_ptrn.alpha=alpha;
rad_ptrn.Ki=Ki;
rad_ptrn.create='rad_pattern';
```

_____ rad_ptrn.array.m _____

```
function [rad_ptrn]=rad_ptrn_array(array,weights,alpha_range,Ki_range,optional)

%this function outputs the received weighted sum of Voc at each array element
%for a source with unit input.  'rad_ptrn' contains fields 'Voc', 'alpha'
%and 'Ki'.
%weigts is the weigted sum of the Voc's in the array and is a column vector
%y=W'*X
%alpha_range is a vector that contains[alpha_start,alpha_end,alpha_N]
%Ki_range is a vector that contains[Ki_start,Ki_end,Ki_N]
%optional.polar_source use polarization for source coupling
%                 default is 'yes'
%optional.pathloss_source use pathloss for source coupling
%                 default is 'yes'

optional_focus.polar_source='yes';
optional_focus.pathloss_source='yes';

%initalize default farfield source
middle=round(array.N_c/2);
[nfd_antenna]=pull_antenna(array,middle,'nonfeed');
[le_nfd,ro_nfd]=feval(nfd_antenna.name,0,0,nfd_antenna.polar);
J_sph_nfd=xyz2sph(0,0);
J_gnfd=nfd_antenna.local;
ro_source=J_gnfd'*J_sph_nfd'*ro_nfd;
polar=xyz2sph(0,0)*ro_source;
%

if nargin=5
  if isfield(optional,'polar_source')
    optional_focus.polar_source=optional.polar_source;
  end
  if isfield(optional,'pathloss_source')
    optional_focus.pathloss_source=optional.pathloss_source;
  end
end

alpha=[alpha_range(1):(alpha_range(2)-alpha_range(1))/(alpha_range(3)-1):alpha_range(3)-1):....
  alpha_range(2)];
Ki=[Ki_range(1):(Ki_range(2)-Ki_range(1))/(Ki_range(3)-1):Ki_range(2)];

for k=1:alpha_range(3)
  comments([['Radiation Pattern ',num2str(k/alpha_range(3),2)]);
  for l=1:Ki_range(3)
    if alpha(k)==0 & Ki(1)==0
      Voc(k,1)=Voc(1,1);
    else
      [source_ki]=build_far(alpha(k),Ki(1),0,1,polar);
      [Voc_ki]=array_focus(array,source_ki,optional_focus);
      Voc(k,1)=weights'*Voc_ki;
    end
```

```
end

end

rad_ptrn.Voc=Voc;
rad_ptrn.alpha=alpha;
rad_ptrn.ki=Ki;
rad_ptrn.create='rad_pattern';
```

_____ QOL.m _____

```
function [lens]=QOL(shape,deminsion,antenna,lensing,delay_0,theta_0,F,unitcellsize,lattice)

%[lens]=QOL(shape,dimension,antenna,lensing,delay_0,theta_0,F,unitcellsize,lat
%Shape is a string argument that describes the lens shape 'circle','rect'
%dimension is a vector that contains the important deminsions for the lens
% circle [radius]
% rect [width,height] width = x dimension and height = y dimension
%antenna is cell discribes the feed and non-feed side antenna types,
%polarization and orientation.  The polarization vector is for a model
%antenna in the oreintation for the theorical or measured radiation patterns.
%Orientation discribes the rotation of the QOL antennas from the model antenna.
%The information is listed as to cells in the order of {feed side,
%non-feedside}, where feed side is a sub cell containing {antenna,
%polarization,orientation}
% polarization} %lensing is a string that discribes the in how many dimensions the len
%focuses in
% ['2D'], ['1Dx'], ['1Dy']
%delay_0 is the base delay for the unitcell in the center of the lens
%unitcellsize defines the x/y dimensions for unitcell
% [width,height] width = x  height = y
%latice is a cell of strings that discribes the the how the unitcells are
%placed in the lens {pattern,boundary}
% PATTERN
% 'triangle' uses a triangular lattice to fill the shape trying to maximise
% the number of unit cells
% 'max' uses a row structure with as many unitcells as possible in
% each row
% BOUNDARY
% 'best' is the tightest packing allowing for 1/4 of the unitcell
% to extend outside the defined shape
% 'limit' is the tightest packing without letting the unitcells
% extend outside the defined shape
%note: all dimensions are in wave length of the center frequency
%note: this code assumes for now that the lens is confined to a plane

%initialize lens
```

```matlab
optional_focus.pathloss_feed='yes';

if nargin == 4
  if isfield(optional,'polar_source')
    optional_focus.polar_source=optional.polar_source;
  end
  if isfield(optional,'polar_feed')
    optional_focus.polar_feed=optional.polar_feed;
  end
  if isfield(optional,'pathloss_source')
    optional_focus.pathloss_source=optional.pathloss_source;
  end
  if isfield(optional,'pathloss_feed')
    optional_focus.pathloss_feed=optional.pathloss_feed;
  end
  if isfield(optional,'mutual')
    mutual=optional.mutual;
  end
end


for k=1:length(channel)
  for l=1:length(imager)
    %comments([['QOLTF',num2str(((k-1)*length(channel)+1)/(length(channel)*...
    %  length(imager)),3)]);

    source=channel{k};

    detector=imager{l};
    [QOL_TF(1,k),Iin_fd_K]=lens_focus(lens,source,detector,optional_focus);

    if ~strcmp(mutual.use,'no')
      [mutual_Z]=get_mutual_Z(lens,mutual);
      switch(mutual.use)
        case {'yes'}
          Iin_fd_mutual_k=mutual_Z*Iin_fd_K;
        case {'max'}
          Iin_fd_mutual_K=abs(mutual_Z)*abs(Iin_fd_K);
      end
      [mutual_TF(1,k)]=mutual_focus(lens,Iin_fd_mutual_K,detector,...
        optional_focus);
    end
  end
end
```

```matlab
lens.device='lens';
lens.shape=shape;
lens.dimension=dimension;
lens.cell.width=unitcellsize(1);
lens.cell.height=unitcellsize(2);
lens.delay_0=delay_0;
lens.theta_0=theta_0;
lens.F=F;
lens.lensing=lensing;
switch shape
  case {'circle'}
    [lens]=packcircle(latice,lens);
  case {'rect'}
    [lens]=packrect(latice,lens);
end

lens.N_c=sum(lens.R);

%determine non-feed antenna points and delays

[lens]=fdside(lensing,lens);
[lens]=packant(lens,antenna);
lens.N_r=length(lens.R);
```

_____ QOLTF.m _____

```matlab
function [QOL_TF,mutual_TF]=QOLTF(lens,channel,imager,optional)

%this function creates a tranfer function for channel to imager
%mutual_TF is the transfer function produced by the mutual coupling currents
%optional.polar_source use polarization for source coupling
%                      default is 'yes'
%optional.polar_feed use polarization for feed coupling
%                    default is 'yes'
%optional.pathloss_source use pathloss for source coupling
%                         default is 'yes'
%optional.pathloss_feed use pathloss for feed coupling
%                       default is 'yes'
%optional.mutual.use ('yes','no','max') calculates the image do to mutual
%                coupling. 'yes' uses the estimated magnitude and phase
%                of the coupled currents and 'max' uses just the magnitude
%                default is 'no'
%optional.mutual.extra is an open field for use by name_mutual.m mutual
%coupling calculations functions

mutual.use='no';
optional_focus.polar_source='yes';
optional_focus.polar_feed='yes';
optional_focus.pathloss_source='yes';
```

```
%theta_r is local coordinat rotation
%polar is local sphereical corridaint polarization

antenna.pos=[pos];
antenna.name=[name];
antenna.base=[get_base(a_z)];
antenna.theta_r=[theta_r];
antenna.local=[rotate_base(antenna.base(1),antenna.theta_r(1))];
antenna.polar=[polar];
```

_____ build_det.m _____

```
function [detector]=build_det(name,pos,theta_r,g,cell_g,width,height,delay,polar)

%This function makes a detector that point to the center of the lens

a_z=(-1*pos')/sqrt(pos'*pos);
[detector.cell]=build_ant(name,pos,a_z,theta_r,polar);
detector.g=g;
detector.device='detector';
detector.N_c=1;
detector.cell.g=[cell_g];
detector.cell.width=width;
detector.cell.height=height;
detector.cell.delay=[delay];
```

_____ build_far.m _____

```
function [source_far]=build_far(alpha,Ki,theta_r,field,polar)

%this function makes a far field source receive at angle 'alpha'
%and 'Ki'. Theta_r is the angle of rotation of local corridiants
%polar is a length 3 column vector discribing the local polarization
%in sphereical cordiants.

source_far.device='source';
source_far.type='farfield';
source_far.field=field;
source_far.angle=[alpha,Ki];
a_z_b=[sin(alpha)*cos(Ki),sin(alpha)*sin(Ki),cos(alpha)];
source_far.base=get_base(a_z_b);
source_far.theta_r=theta_r;
source_far.local=rotate_base(source_far.base,theta_r);
source_far.polar=polar;
source_far.pos=-6*a_z_b';
```

Numerical Model Support Code
_____ array_focus.m _____

```
function [Voc]=array_focus(array,source,optional)

%this fucntion calculates the open circuit voltage at each element of an array
%Voc is a column vector
%optional.pathloss_source defaults is 'yes'
%optional.polar_source defaults is 'yes'

Voc=0;
optional_source.pathloss='yes';
optional_source.polar='yes';

if nargin==3
    if isfield(optional,'pathloss_source')
        optional_source.pathloss=optional.pathloss_source;
    end
    if isfield(optional,'polar_source')
        optional_source.polar=optional.polar_source;
    end
end

for k=1:array.N_c
    [nfd_antenna]=pull_antenna(array,k,'nonfeed');

    switch source.type
        case ('farfield')
            [Voc(k,1)]=couple_farfield(source,nfd_antenna,optional_source);
        case ('nearfield')
            [Voc(k,1)]=couple(source,nfd_antenna,source.lin,optional_source);
    end
end
```

_____ build_ant.m _____

```
function [antenna]=build_ant(name,pos,a_z,theta_r,polar)

%this function creates an antenna element
%pos in global cooridaint
%name is antenna name
%a_z is z unit vector for base corridants (row)
```

─────────────── build_signal.m ───────────────

```
function [signal]=build_signal(channel,constell,w_RF,T_data,user)

%this function builds a signal

signal.channel=channel;
signal.constell=constell;
signal.w_RF=w_RF;
signal.T_data=T_data;
signal.user=user;
```

─────────────── comments.m ───────────────

```
function [Hfig]=comments(text_input)

%this function displays the text in a figure

figure(10)
Hfig=figure(10);
%set(Hfig,'Position',[550,500,300,100]);
menubar=get(Hfig,'Menubar');
switch(menubar)
    case('figure')
        set(Hfig,'Menubar','none');
        axis([-1,1,-1,-1,1,1]);
        text(-2,0,text_input);
        Haxis=gca;
        set(Haxis,'Visible','off');
        Htext=get(Haxis,'Children');
        set(Htext,'FontSize',22);
    case('none')
        Haxis=gca;
        Htext=get(Haxis,'Children');
        set(Htext,'String',text_input);
end
```

─────────────── couple.m ───────────────

```
function [Voc]=couple(transmit,receive,Iin,optional)

%function [Voc]=couple(transmit,receive,Iin,optional)
%this function calculates the open circuit voltage genterated in the
%'receive' antenna by a 'transmit' antenna.
%optional.pathloss default is 'yes'
```

```
%optional.polar default is 'yes'

polar='yes';
optional_Ei.pathloss='yes';

if nargin==4
    if isfield(optional,'polar')
        polar=optional.polar;
    end
    if isfield(optional,'pathloss')
        optional_Ei.pathloss=optional.pathloss;
    end
end

[Ei_mag,ro_t_l]=get_Ei(transmit,receive,Iin,optional_Ei);

[theta_rec,phi_rec]=get_direction(receive,transmit);
[le_r,ro_r_l]=feval(receive.name,theta_rec,phi_rec,receive.polar);
switch polar
    case ('yes')
        Voc=le_r*Ei_mag*ro_r_l'*ro_t_l;
    case ('no')
        Voc=le_r*Ei_mag;
end
```

─────────────── couple.m ───────────────

```
function [Voc]=couple_farfield(source,receive,optional)

%this function calculates the coupling between a farfield source and
% a non-feed side antenna element

polar='yes';

if nargin==3
    if isfield(optional,'polar')
        polar=optional.polar;
    end
end

alpha=source.angle(1);
Ki=source.angle(2);
a_rad=[-1*sin(alpha)*cos(Ki);-1*sin(alpha)*sin(Ki);-1*cos(alpha)];
delay=-dot(a_rad,receive.pos);

Ei_mag=source.field*exp(-j*2*pi*delay);

source.pos=receive.pos+a_rad;
[theta_tran,phi_tran]=get_direction(source,receive);
[theta_rec,phi_rec]=get_direction(receive,source);
```

```
function drawpolygon(local,pos,radius,sides,color)

%drawpolygon(local,pos,radius,sides,color)
%this is a private function
%this function draws a polygon aperture with local coordinates system 'local'
% and position 'pos' in global coordinates

hold on

theta=[0:2*pi/sides:2*pi];
point=[];
for k=1:length(theta)
  xr=radius*cos(theta(k));
  yr=radius*sin(theta(k));
  point=[point,local'*[x;y;0]+pos];
end

x=point(1,:);
y=point(2,:);
z=point(3,:);
plot3(x,-1*z,y,color);
hold off
```

———————————— drawrect.m ————————————

```
function drawrect(local,pos,height,width,color)

%drawrect(local,pos,height,width,color)
%this is a private function
%this function draws a rectangle aperture with local coordinates system 'local'
% and position 'pos' in global coordinates

hold on

h=height;
w=width;

point1=local'*[.5*w;.5*h;0]+ pos;
point2=local'*[-.5*w;.5*h;0]+ pos;
point3=local'*[-.5*w;-.5*h;0]+ pos;
point4=local'*[.5*w;-.5*h;0]+ pos;
point5=local'*[.5*w;.5*h;0]+ pos;

x=[point1(1),point2(1),point3(1),point4(1),point5(1)];
y=[point1(2),point2(2),point3(2),point4(2),point5(2)];
z=[point1(3),point2(3),point3(3),point4(3),point5(3)];
plot3(x,-1*z,y,color);
hold off
```

```
J_sph_tran=xyz2sph(theta_tran,phi_tran);
J_sph_rec=xyz2sph(theta_rec,phi_rec);
J_gt=source.local;
J_gr=receive.local;

ro_t_l=J_sph_rec*J_gr*J_gt'*J_sph_tran'*source.polar;
[le,r,ro_r_l]=feval(receive.name,theta_rec,phi_rec,receive.polar);

switch polar
  case ('yes')
      Voc=le_r*El_mag*ro_r_l'*ro_t_l;
  case ('no')
      Voc=le_r*El_mag;
end
```

———————————— couple_farfield.m ————————————

```
function drawball(radius)

%this function draws a hemisphere of radius 'radius'

figure(7)
view(3)

theta=[0:pi/30:pi/2];
phi=[0:pi/10:2*pi];

for k=1:length(theta)
  for l=1:length(phi)
    ball(k,l)=radius*cos(theta(k));
    x(k,1)=radius*sin(theta(k))*cos(phi(1));
    y(k,1)=radius*sin(theta(k))*sin(phi(1));
  end
end
surf(x,y,ball)

xlabel('x')
ylabel('y')
zlabel('z')

axis('equal')
```

———————————— drawpolygon.m ————————————

```matlab
%initalize variables
lens.cell.fd.pos=lens.cell.nfd.pos;
lens.cell.delay=[];
cell_rad_old=0;

for m=1:lens.N_c
    if lens.cell.nfd.pos{m}(1)==0 & lens.cell.nfd.pos{m}(2) ==0
        x_temp=0;
        y_temp=0;
    else
        cell_rad=sqrt(lens.cell.nfd.pos{m}'*lens.cell.nfd.pos{m});
        temp_rad=cell_rad*sqrt((lens.F^2-(cell_rad^2)*sin(lens.theta_0)^2)...
            /(lens.F^2-cell_rad^2));
        switch lensing
            case {'1Dx'}
                x_temp=(temp_rad/cell_rad)*lens.cell.nfd.pos{m}(1);
                y_temp=lens.cell.nfd.pos{m}(2);
                temp_rad=x_temp;
            case {'1Dy'}
                y_temp=(temp_rad/cell_rad)*lens.cell.nfd.pos{m}(2);
                x_temp=lens.cell.nfd.pos{m}(1);
                temp_rad=y_temp;
            case {'2D'}
                x_temp=(temp_rad/cell_rad)*lens.cell.nfd.pos{m}(1);
                y_temp=(temp_rad/cell_rad)*lens.cell.nfd.pos{m}(2);
        end
    end

    lens.cell.fd.pos{m}(1)=x_temp;
    lens.cell.fd.pos{m}(2)=y_temp;
    %calculate delays
    temp=lens.F*lens.delay_0-0.5*sqrt(lens.F^2+temp_rad^2-2*temp_rad*lens.F ...
        *sin(lens.theta_0));
    temp2=0.5*sqrt(lens.F^2+temp_rad^2+2*temp_rad*lens.F*sin(lens.theta_0));

    lens.cell.delay=[lens.cell.delay,temp+temp2];

    if imag(temp+temp2)>.01
        disp('bad lens design, theta_0 or F is too small')
    end

    %find max radius
    if cell_rad>cell_rad_old
        cell_rad_old=cell_rad;
    end

end

lens.cell.g=ones(1,lens.N_c);
lens.r_max=cell_rad_old;
```

```
_____ drawunit.m _____
```

```matlab
function drawunit(local,pos,color,optional)

%drawunit
%this function plots the local unit vectors at 1/4 length
%optional.font_size allows the change in font size
%     default is 10
%optional.length  length of unit vector
%     default is 0.25

font_size=10;
length_luv=0.25;

if nargin==4
    if isfield(optional,'font_size')
        font_size=optional.font_size;
    end
    if isfield(optional,'length')
        length_luv=optional.length;
    end
end

hold on
xpoint=local'*length_luv*[1; 0; 0] + pos;
ypoint=local'*length_luv*[0; 1; 0] + pos;
zpoint=local'*length_luv*[0; 0; 1] + pos;

plot3([pos(1),xpoint(1)],-1*[pos(3),xpoint(3)],[pos(2),xpoint(2)],color)
plot3([pos(1),ypoint(1)],-1*[pos(3),ypoint(3)],[pos(2),ypoint(2)],color)
plot3([pos(1),zpoint(1)],-1*[pos(3),zpoint(3)],[pos(2),zpoint(2)],color)

[H]=text(xpoint(1),-1*xpoint(3),xpoint(2),'x');
set(H,'FontSize',font_size)
[H]=text(ypoint(1),-1*ypoint(3),ypoint(2),'y');
set(H,'FontSize',font_size)
[H]=text(zpoint(1),-1*zpoint(3),zpoint(2),'z');
set(H,'FontSize',font_size)

hold off
```

```
_____ fdside.m _____
```

```matlab
function [lens]=fdside(lensing,lens)

%[lens]=fdside(lensing,lens)
%This is a private function that calculates the feed side antennas and
%delay in radians base on a paper by McGrath
```

_____ get_base.m _____

```
function [base]=get_base(a_z_b)

%[base]=get_base(a_z_b)
%this function gets to the base corridant system for an antenna with normal
%a_z_b with a_x_b normal a_y and in positive x direction

a_x=[1 0 0];
a_y=[0 1 0];
a_z=[0 0 1];

x_b=cross(a_y,a_z_b);

if x_b*a_x' < 0
    x_b=-1*x_b;
end

a_x_b=x_b/sqrt(x_b*x_b');
a_y_b=cross(a_z_b,a_x_b);

base=[a_x_b;a_y_b;a_z_b];
```

_____ get_direction.m _____

```
function [theta,phi]=get_direction(transmit,receive)

%function [theta,phi]=get_direction(transmit,receive)
%this function calculates the look direction in the local cooridants of transmit
% antenna 'transmit' as it radiates to a receive antenna 'receive'
%'receive' and 'transmit' contain the fields [name,polar,base,local,pos,theta_r]

Rt=transmit.pos;
Rr=receive.pos;
r=Rr-Rt;

r_l=transmit.local*r;
theta=atan2(sqrt(r_l(1)^2+r_l(2)^2),r_l(3));

if theta <0
    theta = theta + 2*pi;
end

phi=atan2(r_l(2),r_l(1));

if phi <0
    phi=phi+2*pi;
end
```

_____ get_Ei.m _____

```
function [Ei_mag,ro_t_l]=get_Ei(transmit,receive,Iin,optional)

%function [Ei_mag,ro_t_l]=get_Ei(transmit,receive,Iin,optional)
%this function calcualtes the incident electric field on a recieve antenna
%in the local cooridainats of the recieve antenna keeping polarization separte.
%It assumes unit input current and the output is a vector of the electric
%field in the local spherical cooridants [r,theta,phi]. There is an
%optional input that allows ignoring 1/r loss. optional.pathloss default
%is 'yes'.
%'receive' and 'transmit' contain the fields [name,polar,base,local,pos,theta_r]

pathloss='yes';
if nargin==4
    if isfield(optional,'pathloss')
        pathloss=optional.pathloss;
    end
end

Rt=transmit.pos;
Rr=receive.pos;
r=Rr-Rt;
r_mag=sqrt(r'*r);

[theta_tran,phi_tran]=get_direction(transmit,receive);
[theta_rec,phi_rec]=get_direction(receive,transmit);

[le_t,ro_t]=feval(transmit.name,theta_tran,phi_tran,transmit.polar);

J_sph_tran=xyz2sph(theta_tran,phi_tran);
J_sph_rec=xyz2sph(theta_rec,phi_rec);
J_gt=transmit.local;
J_gr=receive.local;

ro_t_l=J_sph_rec*J_gr*J_gt'*J_sph_tran'*ro_t;
switch pathloss
    case('no')
        Ei_mag=377*((-j*2*pi)/(4*pi))*le_t*exp(-j*2*pi*r_mag)*Iin;
    case('yes')
        Ei_mag=377*((-j*2*pi)/(4*pi*r_mag))*le_t*exp(-j*2*pi*r_mag)*Iin;
end
```

```
%this fuction calculates the open circuit voltage for a detectot
%do to a source focus through a lens.
%optional.pathloss defaults is 'yes'
%optional.polar defaults is 'yes'
%Iin_fd is the input current to each feed side antena for use in mutual
%coupling calculations

Voc=0;
optional_source.pathloss='yes';
optional_source.polar='yes';
optional_feed.pathloss='yes';
optional_feed.polar='yes';

if nargin==4
  if isfield(optional,'pathloss_feed')
    optional_feed.pathloss=optional.pathloss_feed;
  end
  if isfield(optional,'pathloss_source')
    optional_source.pathloss=optional.pathloss_source;
  end
  if isfield(optional,'polar_feed')
    optional_feed.polar=optional.polar_feed;
  end
  if isfield(optional,'polar_source')
    optional_source.polar=optional.polar_source;
  end
end

for k=1:lens.N_c
  [nfd_antenna]=pull_antenna(lens,k,'nonfeed');

  switch source.type
    case ('farfield')
      [Voc_nfd]=couple_farfield(source,nfd_antenna,optional_source);
    case ('nearfield')
      [Voc_nfd]=couple(source,nfd_antenna,source.Iin,optional_source);
  end

  Iin_fd(k,1)=(Voc_nfd/2)*lens.cell.g(k)*exp(-j*2*pi*lens.cell.delay(k));
  [fd_antenna]=pull_antenna(lens,k,'feed');

  for l=1:detector.N_c
    [det_antenna]=pull_antenna(detector,l);
    [Voc_l]=couple(fd_antenna,det_antenna,Iin_fd(K),optional_feed);
    Voc=Voc+Voc_l*detector.cell.g(l)*exp(-j*2*pi*detector.cell.delay(l))*....
      detector.g;
  end
end
```

```
%define phi=0 hen theta is 0 or pi
if theta < 10e-4 | abs(theta-pi) < 10e-4
  phi=0;
end
_____ get_focal.m _____

function [focal_rad]=get_focal(lens, theta)

%this function calculates the focal distance for a 'lens' at angle theta
%this focal distance is really only known for circle antennas

theta=abs(theta);
phi=asin(lens.r_max/(lens.F));
temp=lens.F*(1+0.5*((sin(phi))^2*sin(theta)^2)/((1-sec(phi)) ...
  *(1+sin(phi)*sin(theta)))));
focal_rad=sec(lens.theta_0)*temp;
_____ get_mutual_Z.m _____

function [mutual_Z]=get_mutual_Z(lens,mutual)

%this function create the coupling matrix between feed side antennas
%mutual.extra is an open field for uses with in mutual coupling
%calculating functions

mutual_Z=zeros(lens.N_c,lens.N_c);
for k=1:lens.N_c
  for l=1:lens.N_c
    if k==l
      mutual_couple(k,1)=0;
    else
      r_21_g=lens.cell.fd.pos{k}-lens.cell.fd.pos{l};
      r_21_l=lens.cell.fd.local{K}*r_21_g;
      temp=lens.cell.fd.name{K};
      space=findstr('_',temp);
      func_name=[temp(1:space-1),'_mutual'];
      [Z_21]=feval(func_name,r_21_l,mutual.extra);
      mutual_Z(k,1)=Z_21;
    end
  end
end
_____ lens_focus.m _____

function [Voc,Iin_fd]=lens_focus(lens,source,detector,optional)
```

---
mutual_focus.m

```
function [Voc]=mutual_focus(lens,Iin_fd_mutual,detector,optional)

%this funcition calculates the open circuit voltage at a detector
%do to mutual coupling currents
%optional.polar_feed 'yes','no' to use polarization between feed and lens
%                 default is 'yes'
%optional.pathloss_feed 'yes','no' to use pathloss between feed and lens
%                 default is 'yes'

Voc=0;
optional_feed.polar='yes';
optional_feed.pathloss='yes';

if nargin==4
  if isfield(optional,'polar_feed')
    optional_feed.polar=optional.polar_feed;
  end

  if isfield(optional,'pathloss_feed')
    optional_feed.pathloss=optional.pathloss_feed;
  end
end

for k=1:lens.N_c
  for l=1:detector.N_c
    [fd_antenna]=pull_antenna(lens,k,'feed');
    [det_antenna]=pull_antenna(detector,l);
    [Voc_1]=couple(fd_antenna,det_antenna,Iin_fd_mutual(k),...
      optional_feed);
    Voc=Voc+Voc_1*detector.cell.g(l)*detector.g*...
      exp(-j*2*pi*detector.cell.delay(l));
  end
end
```

---
omni_antenna.m

```
function [le_t,ro_t]=omni_antenna(theta,phi,polar)

%function [le_t,ro_t]=omni_antenna(theta,phi,polar)
%this function returns the magnitude of the effective length and polarization
%for a omni directional in local coordiants at angels (theta,phi)
%The structure 'polar' contains the polarization of the omin directional
%antenna in local xyz coordiants the default is x-polarized.

if isempty(polar);
```

```
  polar=[0;1;0];
end

J0=sqrt(1/(377*pi));
J_sph=xyz2sph(theta,phi);
if abs(theta-pi/2)> 10e-3
  ro_t=[0 0 0;0 1 0;0 0 1]*J_sph*polar;
  ro_t=ro_t/sqrt(ro_t'*ro_t) ;
  le_t=J0;
else
  ro_t=[0;0;0];
  le_t=0;
end
```

---
omni_mutual.m

```
function [Z_21]=omni_mutual(r_21,extra)

%this function calculates the complex impedance between to omni antennas
%extra contains the ratio of free space wave length  to effective
%wave length in material

%update with papers

Z_21_mag=1/(r_21'*r_21);
Z_21_phs=exp(-j*2*pi*sqrt(r_21'*r_21)*extra)
```

---
packant.m

```
function [lens]=packant(lens,antenna)

%[lens]=packant(lens,antenna)
%This function orients the antennas in the lens.  It assumes the antenna
%normals to point away from the lens.

a_x=[1 0 0];
a_y=[0 1 0];
a_z=[0 0 1];

for m=1:lens.N_c
  lens.cell.nfd.name(m)=antenna.nfd.name;
  lens.cell.nfd.polar(m)=antenna.nfd.polar;
  lens.cell.nfd.theta_r(m)=antenna.nfd.theta_r;
  lens.cell.nfd.base(m)=get_base([0 0 -1]);
  lens.cell.nfd.local(m)=rotate_base(lens.cell.nfd.base(m),antenna.nfd.theta_r);
```

```
lens.cell.fd.name{m}=antenna.fd.name;
lens.cell.fd.polar{m}=antenna.fd.polar;
lens.cell.fd.theta_r{m}=antenna.fd.theta_r;
lens.cell.fd.base{m}=get_base([0 0 1]);
lens.cell.fd.local{m}=rotate_base(lens.cell.fd.base{m},antenna.fd.theta_r);

end

_____ packant2.m _____

function [array]=packant2(array,antenna)

%[array]=packant2(array,antenna)
%This function orients the antennas in the array.  It assumes the antenna
%normals to point away from the array.

a_x=[1 0 0];
a_y=[0 1 0];
a_z=[0 0 1];

for m=1:array.N_c
    array.cell.nfd.name{m}=antenna.nfd.name;
    array.cell.nfd.polar{m}=antenna.nfd.polar;
    array.cell.nfd.theta_r{m}=antenna.nfd.theta_r;
    array.cell.nfd.base{m}=get_base([0 0 -1]);
    array.cell.nfd.local{m}=rotate_base(array.cell.nfd.base{m},...
        antenna.nfd.theta_r);

end

_____ packcircle.m _____

function [lens]=packcircle(latice,lens)

%function [lens]=packcircle(dimensionunitcellsize,latice)
%this is a private function that packs the unitcells into a circle
%based on latice conditions

%initialize variables
rad=lens.dimension(1);
lens.R=[];
```

```
lens.cell.nfd.pos=[];
x_store=[];
y_store=[];
z_store=[];

switch latice{1}
    case {'max'}

    switch latice{2}
        case {'best'}
            lens.N_r=round(2*rad/lens.cell.height);
        case {'limit'}
            lens.N_r=floor(2*rad/lens.cell.height);
    end

    temp=[1:lens.N_r];
    temp2=((lens.N_r/2)+0.5)*ones(1,lens.N_r);
    temp3=lens.cell.height*(temp-temp2);
    y_temp=temp3([lens.N_r:-1:1]);

    for m=1:lens.N_r;
        row_length=2*sqrt((rad)^2-(y_temp(m))^2);
        switch char(latice(2))
            case {'best'}
                N_cell=round(row_length/lens.cell.width);
            case {'limit'}
                N_cell=floor(row_length/lens.cell.width);
        end

        temp=[1:N_cell];
        temp2=((N_cell/2)+0.5)*ones(1,N_cell);
        x_temp=lens.cell.width*(temp-temp2);

        lens.R=[lens.R,N_cell];
        x_store=[x_store,x_temp];
        y_store=[y_store,y_temp(m)*ones(1,N_cell)];
        z_store=[z_store,y_temp(m)*zeros(1,N_cell)];

    end

    for m=1:length(x_store)
        lens.cell.nfd.pos{m}=[x_store(m);y_store(m);z_store(m)];
    end

    case {'triangle'}

    switch latice{2}
        case {'best'}
            lens.N_r=round(2*rad/lens.cell.height);
        case {'limit'}
```

```
        lens.N_r=floor(2*rad/lens.cell.height);
end

    case {'limit'}
        N_cell1=floor(((row_length/2)+(-offset+0.5)*lens.cell.width) ...
            /lens.cell.width);
        N_cell2=floor(((row_length/2)+(offset-0.5)*lens.cell.width) ...
            /lens.cell.width);
        N_cell=N_cell1+N_cell2;
end

temp=[1:N_cell];
temp2=((N_cell2+1 )-offset)*ones(1,N_cell);
x_temp=lens.cell.width*(temp-temp2);

lens.R=[lens.R,N_cell];
x_store=[x_store,x_temp];
y_store=[y_store,y_temp(m)*ones(1,N_cell)];
z_store=[z_store,y_temp(m)*zeros(1,N_cell)];

offset=mod(offset+0.5,1);
end
for m=1:length(x_store)
    lens.cell.nfd.pos(m)=[x_store(m);y_store(m);z_store(m)];
end

end
```

```
_____ packrect.m _____

function [lens]=packrect(latice,lens)

% [lens]=packrect(dimension,unitcellsize,latice)
%this is a private function that packs the unitcells into a rectangle
%based on latice conditions

%initalize variables
width=lens.dimension(1);
height=lens.dimension(2);
lens.R=[];
x_store=[];
y_store=[];
z_store=[];

switch latice{1}

    case {'max'}

        switch latice{2}
```

```
temp=[1:lens.N_r];
temp2=((lens.N_r/2)+0.5)*ones(1,lens.N_r);
temp3=lens.cell.height*(temp-temp2);
y_temp=temp3([lens.N_r:-1:1]);

%find offset at row one
mid_row=round(lens.N_r/2);
row_length=2*sqrt((rad)^2-y_temp(mid_row)^2);

switch latice{2}
    case {'best'}
        N_cell=round(row_length/lens.cell.width);
    case {'limit'}
        N_cell=floor(row_length/lens.cell.width);
end

if rem(N_cell,2)==0
    mid_offset=0.5;
else
    mid_offset=0;
end

offset=mod((mid_offset+(mid_row-1)*0.5),1);

for m=1:lens.N_r;
    row_length=2*sqrt((rad)^2-(y_temp(m))^2);

    switch latice{2}
        case {'best'}
            N_cell1=floor(((row_length/2)+(-offset+0.5)*lens.cell.width) ...
                /lens.cell.width);
            if rem(((row_length/2)+(-offset+0.5)*lens.cell.width), ...
                lens.cell.width)> 0.75*lens.cell.width
                N_cell1=N_cell1+1;
            end

            N_cell2=floor(((row_length/2)+(offset-0.5)*lens.cell.width) ...
                /lens.cell.width);
            if rem(((row_length/2)+(offset-0.5)*lens.cell.width), ...
                lens.cell.width)> 0.75*lens.cell.width
                N_cell2=N_cell2+1;
            end

            N_cell=N_cell1+N_cell2;
```

```matlab
case {'best'}
    lens.N_r=round(height/lens.cell.height);
case {'limit'}
    lens.N_r=floor(height/lens.cell.height);
end

temp=[1:lens.N_r];
temp2=((lens.N_r/2)+0.5)*ones(1,lens.N_r);
temp3=lens.cell.height*(temp-temp2);
y_temp=temp3([lens.N_r:-1:1]);

for m=1:lens.N_r;
    switch latice{2}
        case {'best'}
            N_cell=round(width/lens.cell.width);
        case {'limit'}
            N_cell=floor(width/lens.cell.width);
    end

temp=[1:N_cell];
temp2=((N_cell/2)+0.5)*ones(1,N_cell);
x_temp=lens.cell.width*(temp-temp2);

lens.R=[lens.R,N_cell];
x_store=[x_store,x_temp];
y_store=[y_store,y_temp(m)*ones(1,N_cell)];
z_store=[z_store,y_temp(m)*zeros(1,N_cell)];

end

for m=1:length(x_store)
    lens.cell.nfd.pos{m}=[x_store(m);y_store(m);z_store(m)];
end

case {'triangle'}

switch latice{2}
    case {'best'}
        lens.N_r=round(height/lens.cell.height);
    case {'limit'}
        lens.N_r=floor(height/lens.cell.height);
end

temp=[1:lens.N_r];
temp2=((lens.N_r/2)+0.5)*ones(1,lens.N_r);
temp3=lens.cell.height*(temp-temp2);
y_temp=temp3([lens.N_r:-1:1]);

%find offset at row one
```

```matlab
mid_row=round(lens.N_r/2);
switch latice{2}
    case {'best'}
        N_cell=round(width/lens.cell.width);
    case {'limit'}
        N_cell=floor(width/lens.cell.width);
end

if rem(N_cell,2)==0
    mid_offset=0.5;
else
    mid_offset=0;
end

offset=mod((mid_offset+(mid_row-1)*0.5),1);

for m=1:lens.N_r;

    switch latice{2}
        case {'best'}
            N_cell1=floor(((width/2)+(-offset+0.5)*lens.cell.width) ...
                /lens.cell.width);

            if rem(((width/2)+(-offset+0.5)*lens.cell.width), ...
                lens.cell.width)> 0.75*lens.cell.width

                N_cell1=N_cell1+1;

            end

            N_cell2=floor(((width/2)+(offset-0.5)*lens.cell.width) ...
                /lens.cell.width);

            if rem(((width/2)+(offset-0.5)*lens.cell.width), ...
                lens.cell.width)> 0.75*lens.cell.width

                N_cell2=N_cell2+1;

            end
            N_cell=N_cell1+N_cell2;
        case {'limit'}
            N_cell1=floor(((width/2)+(-offset+0.5)*lens.cell.width) ...
                /lens.cell.width);
            N_cell2=floor(((width/2)+(offset-0.5)*lens.cell.width) ...
                /lens.cell.width);
            N_cell=N_cell1+N_cell2;

    end

    temp=[1:N_cell];
    temp2=((N_cell2+1)-offset)*ones(1,N_cell);
    x_temp=lens.cell.width*(temp-temp2);
```

```matlab
else
    if isfield(cal.polar,'crosspole') & isfield(polar,'crosspole')
        if cal.polar.crosspole~=polar.crosspole | cal.polar.delta~=polar.delta
            needcal='yes';
        end
    elseif isfield(cal.polar,'crosspole') & ~isfield(polar,'crosspole')
        needcal='yes';
    elseif ~isfield(cal.polar,'crosspole') & isfield(polar,'crosspole')
        needcal='yes';
    end
end
end

switch(needcal)
case('yes')

disp('calibrating patch antenna')

Prad=0;

theta_step=pi/60;
phi_step=pi/20;
theta_temp=[0:theta_step:pi/2];
phi_temp=[0:phi_step:2*pi];

L_mag=[];
flux=0;
for k=1:length(theta_temp)
    for l=1:length(phi_temp)
        L_temp=patch_antenna_L(theta_temp(k),phi_temp(l),polar);
        L_mag(k,l)=L_temp'*L_temp;
    end
end

for k=1:length(theta_temp)-1
    for l=1:length(phi_temp)-1
        L_mag_avg=sum([L_mag(k,l),L_mag(k+1,l),L_mag(k,l+1),....
            L_mag(k+1,l+1)])/4;
        area=sin((theta_temp(k)+theta_step*(k+1))/2)*theta_step*phi_step;
        flux=flux+L_mag_avg*area;
    end
end

M0=1/sqrt(((2*pi)^2/(16*377*pi^2))*flux);
cal.M0=M0;
cal.polar=polar;

Globalcal_patch=cal;
save /home/sydney/vian/matlab/toolbox/Q0lens/working/patch_antenna_cal cal
```

263

```matlab
lens.R=[lens.R,N_cell];
x_store=[x_store,x_temp];
y_store=[y_store,y_temp(m)*ones(1,N_cell)];
z_store=[z_store,y_temp(m)*zeros(1,N_cell)];
offset=mod(offset+0.5,1);
end

for m=1:length(x_store)
    lens.cell.nfd.pos(m)=[x_store(m);y_store(m);z_store(m)];
end

end
```

_____ patch_antenna.m _____

```matlab
function [le_t,ro_t]=patch_antenna(theta,phi,polar)

%function [le_t,ro_t]=patch_antenna(theta,phi,polar)
%this function returns the magnitude of the effective length and polarization
%for a patch in local cooridants at angels (theta,phi).  The structure
%'polar' contains additional informationi specific to this antenna type.
%'polar.Le   is the effective length of patch antenna
%'polar.We   is the effective width of patch antenna
%'polar.h    is the effective heigth of patch antenna
%'polar.crosspole  is the cross polar ratio in dB   default is infinite
%'polar.delta   is the phase shift between the two polarizations (determines
%                RHP LHP)

global Globalcal_patch

needcal='no';
if ~isempty(Globalcal_patch)
    cal=Globalcal_patch;
    verify='yes';
elseif exist('patch_antenna_cal.mat','file')
    disp('reading patch cal from file')
    load('patch_antenna_cal');
    Globalcal_patch=cal;
    verify='yes';
else
    needcal='yes'
    verify='no';
end

if strcmp(verify,'yes')
    if cal.polar.Le~=polar.Le | cal.polar.We~=polar.We | cal.polar.h~=polar.h
        needcal='yes';
```

```
L_theta=cos(theta)*cos(phi)*temp1*temp2*AF;
L_phi=-sin(phi)*temp1*temp2*AF;
L=[0;L_theta;L_phi];

if isfield(polar,'crosspole')
    CR=10^(polar.crosspole/10);

    if isfield(polar,'delta')
        delta=polar.delta;
    else
        delta=0;
    end

    Z=2*pi*polar.h*cos(theta);
    Y=2*pi*(polar.Le/2)*sin(theta)*sin(phi);
    AF=2*cos(2*pi*(polar.We/2)*(sin(theta)*cos(phi)));

    if Z==0
        temp1=1;
    else
        temp1=sin(Z)/Z;
    end

    if Y==0
        temp2=1;
    else
        temp2=sin(Y)/Y;
    end

    L_theta=cos(theta)*sin(phi)*temp1*temp2*AF;
    L_phi=cos(phi)*temp1*temp2*AF;
    L_temp=[0;L_theta;L_phi]*exp(j*delta);

    A=1/(1+CR);
    B=1-A;
    L=B*L+A*L_temp;

end

if theta>pi/2
    L=[0;0;0];
end
```

——————— patch_mutual.m ———————

```
function [Z_21]=patch_mutual(r_21,extra)

%this function calculates the complex impedance between to patch antennas
%extra.lambda_eff  contains the ratio of free space wave length  to effective
```

```
case('no')

    M0=cal.M0;

end

L=patch_antenna_L(theta,phi,polar);
field=(-M0/377)*cross([1;0;0],L);

if field'*field < 10e-9
    le_t=0;
    ro_t=[0;0;0];
else
    le_t=sqrt(field'*field);
    ro_t=field/le_t;
end
```

——————— patch_antenna_L.m ———————

```
function [L]=patch_antenna_L(theta,phi,polar)

%this function calculates the results of the surface integral of the
%magnetics currents.
%'polar' contains additional informationi specific to this antenna type.
%polar.Le  is the effective length of patch antenna
%polar.We  is the effective width of patch antenna
%polar.h   is the effective heigth of patch antenna
%
%polar.crosspole  is the cross polar ratio in dB  default is infinite
%polar.delta   is the phase shift between the two polarizations (determines
%                  RHP LHP)
%note i think i want the polar to contain fields (polar,crosspole,length)

Z=2*pi*polar.h*cos(theta);
X=2*pi*(polar.We/2)*sin(theta)*cos(phi);
AF=2*cos(2*pi*(polar.Le/2)*(sin(theta)*sin(phi)));

if Z==0
    temp1=1;
else
    temp1=sin(Z)/Z;
end

if X==0
    temp2=1;
else
    temp2=sin(X)/X;
end
```

```
        antenna.local=device.cell.nfd.local(antenna_num);
        antenna.pos=device.cell.nfd.pos(antenna_num);
        antenna.theta_r=device.cell.nfd.theta_r(antenna_num);
    case('feed')
        antenna.name=device.cell.fd.name(antenna_num);
        antenna.polar=device.cell.fd.polar(antenna_num);
        antenna.base=device.cell.fd.base(antenna_num);
        antenna.local=device.cell.fd.local(antenna_num);
        antenna.pos=device.cell.fd.pos(antenna_num);
        antenna.theta_r=device.cell.fd.theta_r(antenna_num);
    end
    case('detector')
        antenna.name=device.cell.name(antenna_num);
        antenna.polar=device.cell.polar(antenna_num);
        antenna.base=device.cell.base(antenna_num);
        antenna.local=device.cell.local(antenna_num);
        antenna.pos=device.cell.pos(antenna_num);
        antenna.theta_r=device.cell.theta_r(antenna_num);
end
```

_____ rotate.base.m _____

```
function [local]=rotate_base(base,theta_r)

%[local]=rotate_base(base,theta_r)
%this function rotates the base coordinat system through an angle 'theta_r'
%in the base coordiants system xy plane

rotate=[cos(theta_r) sin(theta_r) 0; -sin(theta_r) cos(theta_r) 0; 0 0 1];
local=rotate*base;
```

_____ semi_antenna.m _____

```
function [le_t,ro_t]=semi_antenna(theta,phi,polar)

%function [le_t,ro_t]=semi_antenna(theta,phi,polar)
%this function returns the magnitude of the effective length and polarization
%for a semi directional in local coordiants at angels (theta,phi)
%The structure 'polar' contains the polarization of the omin directional
%antenna in local xyz coordiants the default is x-polarized.

if isempty(polar);
    polar=[0;1;0];
```

```
%wave length in material
%extra.ro contain the scale constant for the magnitude coupling

%update with papers

theta=atan2(sqrt(r_21(1)^2+r_21(2)^2),r_21(3));

if theta <0
    theta = theta + 2*pi;
end

phi=atan2(r_21(2),r_21(1));

if phi <0
    phi=phi+2*pi;
end

%define phi=0 hen theta is 0 or pi
if theta < 10e-4 | abs(theta-pi) < 10e-4
    phi=0;
end

r_21_mag=sqrt(r_21'*r_21);
Z_21_mag=(extra.ro/r_21_mag)^(1+sin(phi));
Z_21_phs=exp(-j*2*pi*r_21_mag*extra.lambda_eff);
Z_21=Z_21_mag*Z_21_phs;
```

_____ pull_antenna.m _____

```
function [antenna]=pull_antenna(device,antenna_num,lens_side)

%this function pulls out one antenna from a device, the lens_side
%is only needed if the device is a lens and not a detector

switch device.device
    case('array')
        antenna.name=device.cell.nfd.name(antenna_num);
        antenna.polar=device.cell.nfd.polar(antenna_num);
        antenna.base=device.cell.nfd.base(antenna_num);
        antenna.local=device.cell.nfd.local(antenna_num);
        antenna.pos=device.cell.nfd.pos(antenna_num);
        antenna.theta_r=device.cell.nfd.theta_r(antenna_num);
    case('lens')
        switch lens_side
        case('nonfeed')
            antenna.name=device.cell.nfd.name(antenna_num);
            antenna.polar=device.cell.nfd.polar(antenna_num);
            antenna.base=device.cell.nfd.base(antenna_num);
```

```
%this function outputs the receive Voc at a detector defined by the imager
%and weights for a source with unit input.
%'rad_ptrn' contains fields 'Voc', 'alpha' and 'Ki'.
%alpha_range is a vector that contains[alpha_start,alpha_end,alpha_N]
%Ki_range is a vector that contains[Ki_start,Ki_end,Ki_N]
%optional.polar_source use polarization for source coupling
%                   default is 'yes'
%optional.polar_feed use polarization for feed coupling
%                   default is 'yes'
%optional.pathloss_source use pathloss for source coupling
%                   default is 'yes'
%optional.pathloss_feed use pathloss for feed coupling
%                   default is 'yes'
optional_rad.polar_source='yes';
optional_rad.polar_feed='yes';
optional_rad.pathloss_source='yes';
optional_rad.pathloss_feed='yes';
if nargin==6
    optional_rad=optional;
end

for m=1:length(imager)
    m/length(imager)
    if weight(m)~=0
        detector_m=imager{m};
        [rad_ptrn_m]=rad_pattern(lens,detector_m,alpha_range,ki_range...
            ,optional_rad);
        if m==1
            rad_ptrn=rad_ptrn_m;
            rad_ptrn.Voc=weight(m)*rad_ptrn_m.Voc;
        else
            rad_ptrn.Voc=rad_ptrn.Voc+weight(m)'*rad_ptrn_m.Voc;
        end
    end
end

_____ cal.W.est.m _____

function [W_est]=cal_W_est(R_est,P_est,mask)

mask_pos=[];
for k=1:length(mask)
    if mask(k)==1
        mask_pos=[mask_pos,k];
    end
end

R_reduce=R_est(mask_pos,mask_pos);
```

```
end
J0=sqrt(2/(377*pi));
J_sph=xyz2sph(theta,phi);
if theta < (pi/2-10e-3)
    ro_t=[0 0 0;0 1 0;0 0 1]*J_sph*polar;
    ro_t=ro_t/sqrt(ro_t'*ro_t) ;
    le_t=J0;
else
    ro_t=[0;0;0];
    le_t=0;
end

_____ semi.mutual.m _____

function [Z_21]=semi_mutual(r_21,extra)

%this function calculates the complex impedance between to semi antennas
%extra contains the ratio of free space wave length to effective
%wave length in material

%update with papers

Z_21_mag=1/(r_21'*r_21);
Z_21_phs=exp(-j*2*pi*sqrt(r_21'*r_21)*extra)

_____ xyz2sph.m _____

function [J_sph]=xyz2sph(theta,phi)

%function [J_sph]=xyz2sph(theta,phi)
%this function creates an Jacobian to convert cartisian vectors to
%spherical vectors

J_sph=[sin(theta)*cos(phi) sin(theta)*sin(phi) cos(theta); ...
       cos(theta)*cos(phi) cos(theta)*sin(phi) -1*sin(theta); ...
       -1*sin(phi)         cos(phi)            0 ];

_____ LMS Code _____
_____ adapt.rad.ptrn.m _____

function [rad_ptrn]=adapt_rad_ptrn(lens,imager,weight,alpha_range,ki_range,optional)
```

```
R_reduce_inv=inv(R_reduce);
R_sub_inv=zeros(length(mask));
R_sub_inv(mask_pos,mask_pos)=R_reduce_inv;
W_est=R_sub_inv*P_est;
```

---------- csf.m ----------

```
function [signal]=csf(signal,field)

%this function changes the signal field
%field is a vector for each source in the channel

channel_temp=signal.channel;
for k=1:length(field)
  source_temp=channel_temp(k);
  source_temp.field=field(k);
  channel_temp(k)=source_temp;
end

signal.channel=channel_temp;
```

---------- drawvar.m ----------

```
function drawvar(pos,radius,sides,color)

%drawpolygon(local,pos,radius,sides,color)
%this is a private function
%this function draws a polygon aperture with local coordinates system 'local'
% and position 'pos' in global coordinates

hold on

theta=[0:2*pi/sides:2*pi];
point=[];
for k=1:length(theta)
  x=radius*cos(theta(k));
  y=radius*sin(theta(k));
  point=[point,[x;y;0]+pos];
end

x=point(1,:);
y=point(2,:);
z=point(3,:);
plot3(x,y,z,color);
hold off
```

---------- get.data.m ----------

```
function [data_new]=get_data(constell)

%this function randomly detemines the next data sample for a given
%constellation and symbol period.  All constellations have peak power
%of unity.
% support for 'QPSK' 'BPSK' 'carrier'

switch(constell)
  case('QPSK')
    symbols=[1,j,-1,-j];
  case('BPSK')
    symbols=[1, -1]
  case('carrier')
    symbols=[1, 1];
end

data_new=symbols(1+floor(length(symbols)*rand(1,1)));
```

---------- get.data.pwr.m ----------

```
function [data_pwr]=get_data_pwr(constell)

%this function calculates the data pwr
% support for 'QPSK' 'BPSK' 'carrier'

switch(constell)
  case('QPSK')
    symbols=[1,j,-1,-j];
  case('BPSK')
    symbols=[1, -1]
  case('carrier')
    symbols=[1, 1];
end

data_pwr=symbols*symbols'/length(symbols);
```

---------- plot_error.m ----------

```
function plot_error(sub_non,window)

font_size=10;
colors=['y','m','c','r','g','b','y'];
[m,n]=size(sub_non.QOL.e);
```

```
    hold on
        plot(sub_non.array.y_SNR.y_symbol_opt.mean(k),'ro')
        drawpolygon([1 0;0 0 -1;0 1 0],...
            [real(sub_non.array.y_SNR.y_symbol_opt.mean(k));...
            ;0;imag(sub_non.array.y_SNR.y_symbol_opt.mean(k))],...
            sqrt(sub_non.array.y_SNR.y_symbol_opt.var(k)),20,'r')
    hold off
    end
end
if strcmp(item,'QOL')
    if strcmp(type,'normal')
    hold on
    plot(sub_non.QOL.y(pass,[N_y-N_samples+1:N_y]),'b.')
    hold off
    for k=1:length(sub_non.QOL.y_SNR.y_symbol.mean(:,pass))
    hold on
        plot(sub_non.QOL.y_SNR.y_symbol.mean(k,pass),'ro')
        drawpolygon([1 0;0 0 -1;0 1 0],...
            [real(sub_non.QOL.y_SNR.y_symbol.mean(k,pass))],...
            ;0;imag(sub_non.QOL.y_SNR.y_symbol.mean(k,pass))],...
            sqrt(sub_non.QOL.y_SNR.y_symbol.var(k,pass)),20,'r')
    hold off
    end
end
if strcmp(type,'opt')
    hold on
    plot(sub_non.QOL.y_opt(pass,[N_y-N_samples+1:N_y]),'b.')
    hold off
    for k=1:length(sub_non.QOL.y_SNR.y_symbol_opt.mean(:,pass))
    hold on
        plot(sub_non.QOL.y_SNR.y_symbol_opt.mean(k,pass),'ro')
        drawpolygon([1 0;0 0 -1;0 1 0],...
            [real(sub_non.QOL.y_SNR.y_symbol_opt.mean(k,pass))];...
            ;0;imag(sub_non.QOL.y_SNR.y_symbol_opt.mean(k,pass))],...
            sqrt(sub_non.QOL.y_SNR.y_symbol_opt.var(k,pass)),20,'r')
    hold off
    end
end
axis('equal')
%axis('square')
```

_____ plot_eye_para.m _____

```
function plot_eye_para(sub_non,item,type,color)

%this function plots the y, and estimated mean and variance in a eye pattern
%for the item (QOL or QOL) with pass (for QOL it is one)
```

```
for l=1:m
    for k=1:window-1
        e_temp(l,k)=sum(sub_non.QOL.e(l,[1:k]));
    end
    for k=window:n
        e_temp(l,k)=sum(sub_non.QOL.e(l,[k-window+1:k]));
    end
end

for k=1:10:m
    hold on
    plot(abs(e_temp(k,:)),[colors(mod(k,7)+1),'-'])
    [H_temp]=text(n,e_temp(n),num2str(k,3));
    set(H_temp,'FontSize',font_size)
    hold off
end
axis('square')
xlabel('number of signals')
ylabel('error')
```

_____ plot.eye.m _____

```
function plot_eye(sub_non,item,type,pass,N_samples)

%this function plots the y, and estimated mean and variance in a eye pattern
%for the item (QOL or array) with pass (for array it is one)
%type is for normal of opt

clf
N_y=length(sub_non.array.y);
if strcmp(item,'array')
    if strcmp(type,'normal')
    hold on
    plot(sub_non.array.y([N_y-N_samples+1:N_y]),'b.')
    hold off
    for k=1:length(sub_non.array.y_SNR.y_symbol.mean)
    hold on
        plot(sub_non.array.y_SNR.y_symbol.mean(k),'ro')
        drawvar([real(sub_non.array.y_SNR.y_symbol.mean(k))];...
            ;imag(sub_non.array.y_SNR.y_symbol.mean(k));0],...
            sqrt(sub_non.array.y_SNR.y_symbol.var(k)),20,'r')
    hold off
    end
if strcmp(type,'opt')
    hold on
    plot(sub_non.array.y_opt([N_y-N_samples+1:N_y]),'b.')
    hold off
    for k=1:length(sub_non.array.y_SNR.y_symbol_opt.mean)
```

```
%type is for normal of opt

[m,n]=size(sub_non.QOL.y_SNR.y_symbol.mean);
if strcmp(type,'normal')
   for l=[[1:10],n]
      for k=1:m
         hold on
         plot(sub_non.QOL.y_SNR.y_symbol.mean(k,1),[color,'o'])
         drawpolygon([1 0 0;0 -1;0 1 0],...
            [real(sub_non.QOL.y_SNR.y_symbol.mean(k,1)),...
            :0;imag(sub_non.QOL.y_SNR.y_symbol.mean(k,1))],...
            sqrt(sub_non.QOL.y_SNR.y_symbol.var(k,1)),20,color)
         hold off
      end
   end
end
if strcmp(type,'opt')
   for l=1:n
      for k=1:m
         hold on
         plot(sub_non.QOL.y_SNR.y_symbol_opt.mean(k,1),[color,'o'])
         drawpolygon([1 0 0;0 -1;0 1 0],...
            [real(sub_non.QOL.y_SNR.y_symbol_opt.mean(k,1)),...
            :0;imag(sub_non.QOL.y_SNR.y_symbol_opt.mean(k,1))],...
            sqrt(sub_non.QOL.y_SNR.y_symbol_opt.var(k,1)),20,color)
         hold off
      end
   end
end
axis('equal')
```

—————————— plot_signals_pos.m ——————————

```
function plot_signals_pos(signals)
```

%this function plots the signals on the radiation plattern plots
%the numbering is signals,source

```
figure(3)
for k=1:length(signals)
   signal_k=signals{k};
   channel=signal_k.channel
   for l=1:length(channel)
      source_kl=channel{l};

      hold on
      switch (source_Kl.type)
         case('farfield')
            angle_Kl=source_Kl.angle;
```

```
            x=angle_Kl(1)*cos(angle_Kl(2));
            y=angle_Kl(1)*sin(angle_Kl(2));
            plot3(x,y,0,'bx')
            text(x,y,0,[num2str(k,2),'',num2str(1,2)]);
         case('nearfield')
            pos_Kl=source_Kl.pos
            alpha=atan2(sqrt(pos_Kl(1)^2+pos_Kl(2)^2),pos_Kl(3));
            Kl=atan2(pos_Kl(2),pos_Kl(1));
            if Kl<0
               Kl=Kl+2*pi;
            end
            r=angle_Kl(alpha)*cos(angle_Kl(Kl));
            y=angle_Kl(alpha)*sin(angle_Kl(Kl));
            plot3(x,y,0,'bo')
            text(x,y,0,[num2str(k,2),'',num2str(1,2)]);
      end
      hold off
   end
end
```

—————————— plot_W_est_QOL.m ——————————

```
function plot_W_est_QOL(sub_non)
N_imager=121
for k=[1:11]
   mask=zeros(N_imager,1);
   mask(sub_non.QOL.order([1:k]))=ones(k,1);
   R_est=sub_non.QOL.R_est{k};
   P_est=sub_non.QOL.P_est(:,k);
   W_est(:,k)=cal_W_est(R_est,P_est,mask);
end
plot_weights(W_est);
```

—————————— plot_weight_noise.m ——————————

```
function plot_weigh_noise(sub_nons)

font_size=10;
colors=['y','m','c','r','g','b','y'];

figure(1)
clf
for k=1:length(sub_nons)
   hold on
   sub_non=sub_nons{k};
   plot(10*log10(sub_non.QOL.W_noise.mean),[colors(mod(k,7)+1),'-'])
```

```
        clf
        for k=1:m
            hold on
            plot(10*log10(abs(sub_non.QOL.error_min(k,:))),[colors(mod(k,6)+1),'--'])
            hold off
        end
        axis('square')
        xlabel('number of signals')
        ylabel('10log_{10}(|Minimum Error|)')

        figure(4)
        clf
        for k=1:m
            hold on
            plot(10*log10(abs(sub_non.QOL.mu(k,:))),[colors(mod(k,6)+1),'--'])
            hold off
        end
        axis('square')
        xlabel('number of signals')
        ylabel('10log_{10}(mu)')
```

———————————————————————— plot_weights.m ————————————————————————

```
function plot_weights(weights,optional)

%this function plots the weight tracks
%optional.font_size   default is 10
%optional.figure   default is 8

font_size=10;
fig=8;

if nargin==2
    if isfield(optional,'font_size')
        font_size=optional.font_size;
    end
    if isfield(optional,'figure')
        fig=optional.figure;
    end
end

figure(fig)
clf

colors=['y','m','c','r','g','b','y'];
[m,n]=size(weights);

for k=1:m
    hold on
```

```
        hold off
        end
        axis('square')
        xlabel('number of signals')
        ylabel('10log_{10}(weight noise mean)')

        figure(2)
        clf
        for k=1:length(sub_nons)
            hold on
            sub_non=sub_nons{k};
            plot(10*log10(sub_non.QOL.W_noise.var),[colors(mod(k,7)+1),'--'])
            hold off
        end
        axis('square')
        xlabel('number of signals')
        ylabel('10log_{10}(weight noise variance)')
```

———————————————————— plot_weight_noise_cal.m ————————————————————

```
function plot_weigh_noise_cal(sub_non)

font_size=10;
colors=['y','m','c','r','g','b','y'];

[m,n]=size(sub_non.QOL.W_noise.mean);
figure(1)
clf
for k=1:m
    hold on
    plot(10*log10(sub_non.QOL.W_noise.mean(k,:)),[colors(mod(k,6)+1),'--'])
    hold off
end
axis('square')
xlabel('number of signals')
ylabel('10log_{10}(weight noise mean)')

figure(2)
clf
for k=1:m
    hold on
    plot(10*log10(sub_non.QOL.W_noise.var(k,:)),[colors(mod(k,6)+1),'--'])
    hold off
end
axis('square')
xlabel('number of signals')
ylabel('10log_{10}(weight noise variance)')

figure(3)
```

```
                                                sub_LMS_array.m

function [output]=sub_LMS_array(arrayTF_signal,noise_TF,N_samples,signals,time_step,no

%this function does the LMS subset processing on array
%the output contians the weights, y, data, for the system

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%initized variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N_s=length(signals);
[N_array,dumb]=size(noise_TF);
data=zeros(N_s,1);
y=0;
y_opt=0;
W=zeros(N_array,2);
W_opt=zeros(N_array,1);
P_est=zeros(N_array,1);
R_est=zeros(N_array);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%find desire signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
desire_signal=0;
for k=1:N_s
    signal_k=signals(k);
    if strcmp(signal_k.user,'desire')
        desire_signal=k;
    end
end

if desire_signal==0
    disp('no desire user')
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%calculate R,P, Wopt and mu
%assume all signals are uncorrelated
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R=zeros(N_array,N_array);
for k=1:N_s
    signal_k=signals(k);
    TF_signal_k=arrayTF_signal(k);
    [m_k,n_k]=size(TF_signal_k);
    R=R+get_data_pwr(signal_k.constell)*(TF_signal_k*ones(n_k,1))*...
        (TF_signal_k*ones(n_k,1))';
    if k==desire_signal
        P=get_data_pwr(signal_k.constell)*(TF_signal_k*ones(n_k,1));
```

```
    plot(real(weights(k,:)),imag(weights(k,:)),[colors(mod(k,7)+1),','])
    plot(real(weights(k,:)),imag(weights(k,:)),[colors(mod(k,7)+1),':'])
    hold off
end

for k=1:m
    hold on
    plot(real(weights(k,n)),imag(weights(k,n)),'wo')
    [H_temp]=text(real(weights(k,n)),imag(weights(k,n)),num2str(k,3));
    set(H_temp,'FontSize',font_size)
    hold off
end
axis('equal')
xlabel('Re\{W_k\}')
ylabel('Im\{W_k\}')

                                                slicer.m

function [data_guess]=get_data(sample,constell)

%this function performs a minium distance guess of the output data
% support for 'QPSK' 'BPSK' 'carrier'

switch(constell)
case('QPSK')
    symbols=[1,j,-1,-j];
    if angle(sample) <= pi/4 & angle(sample) >= -pi/4
        data_guess=1;
    elseif angle(sample) > pi/4 & angle(sample) < 3*pi/4
        data_guess=j;
    elseif angle(sample) < -pi/4 & angle(sample) > -3*pi/4
        data_guess=-j;
    else
        data_guess=-1;
    end
case('BPSK')
    symbols=[1, -1]
    if real(sample) > 0
        data_guess=1;
    else
        data_guess=-1;
    end
case('carrier')
    symbols=[1, 1];
    data_guess=1;
end

data_new=symbols(1+floor(length(symbols)*rand(1,1)));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%noise
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[m,n]=size(noise_TF);
noise_temp=sqrt(noise_pwr.array)*(randn(n,1)+j*randn(n,1))/sqrt(2);
noise=noise_TF*noise_temp;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LMS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
X=d+noise;
y(1)=W(:,1)'*X;
y_opt(1)=W_opt'*X;
e_l=data(desire_signal,1)-y(1);
e(1)=e_l;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%estimate R
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R_est=R_est+X*X'/N_samples;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

W(:,1+1)=W(:,1)+2*mu*e_l*X;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end
output.data=data;
output.e=e;
output.y=y;
output.y_opt=y_opt;
output.W_opt=W_opt;
output.W=W;
output.R=R;
output.R_est=R_est;
output.P=P;
output.P_est=P_est;


_____ sub_LMS_QOI.m _____

function [output]=sub_LMS_QOI(QOITF_signal,noise_TF,N_imager,N_samples,signals,time_st

%this function does the LMS subset processing on QOI give the mask
%mu is .05/trace[R]
%the output contians the weights, y, data, for the system
```

```
end
end
if noise_pwr.array~=0
R=R*noise_pwr.array*eye(N_array);
end
W_opt=inv(R)*P;
mu=.05/trace(R);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for l=2:N_samples
%comments(['array LMS  ', num2str(1/N_samples,3)]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%get data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
time_now=(l-1)*time_step;
time_then=(l-2)*time_step;
for k=1:N_s
signal_k=signals(k);
if time_then<=floor(time_now/signal_k.T_data)*signal_k.T_data...
& time_now > floor(time_now/signal_k.T_data)*signal_k.T_data
data(k,l)=get_data(signal_k.constell);
else
data(k,l)=data(k,l-1);
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%modulation/demodulation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
signal_temp=signals(desire_signal);
w_IF=signal_temp.w_RF;
d=zeros(N_array,1);
for k=1:N_s
signal_k=signals(k);

TF_signal_k=arrayTF_signal(k);
[m,n]=size(TF_signal_k);
d=d+TF_signal_k*data(k,l)*exp(signal_k.w_RF-w_IF)*...
ones(n,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%estimate P
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if k==desire_signal
P_est=P_est + abs(data(k,l))^2*(TF_signal_k)*...
exp(signal_k.w_RF-w_IF)*ones(n,1)/N_samples;
end
```

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%initized variables
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N_s=length(signals);
data=zeros(N_s,1);
y=0;
y_opt=0;
W=zeros(N_imager,2);
W_opt=zeros(N_imager,1);
P_est=zeros(N_imager,1);
R_est=zeros(N_imager);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%find desire signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
desire_signal=0;
for k=1:N_s
    signal_k=signals(k);

    if strcmp(signal_k.user,'desire')
        desire_signal=k;
    end
end

if desire_signal==0
    disp('no desire user')
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%calculate R,P, W_opt and mu
%assume all signals are uncorrelated
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
R=zeros(N_imager,N_imager);
for k=1:N_s
    signal_k=signals(k);
    TF_signal_k=QOLTF_signal(k);
    [m_k,n_k]=size(TF_signal_k);
    R=R+get_data_pwr(signal_k.constell)*(TF_signal_k*ones(n_k,1)*...
        (TF_signal_k*ones(n_k,1))');
    if k==desire_signal
        P=get_data_pwr(signal_k.constell)*(TF_signal_k*ones(n_k,1));
    end
end
if noise_pwr.lens~=0
    R=R+noise_pwr.lens*noise_TF_QOL.lens*noise_TF_QOL.lens';
end
if noise_pwr.detector~=0
    for k=1:N_imager
```

```matlab
            noise_TF_k=noise_TF_QOL.det(k);
            R_temp(k,k)=noise_pwr.detector*sum(abs(noise_TF_k).^2);
        end
        R=R+R_temp;
    end
    if noise_pwr.imager~=0
        R=R+noise_pwr.imager*eye(N_imager);
    end
    R_sub=diag(mask).'*R*diag(mask);
    mu=.05/trace(R_sub);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%calculating W_opt
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mask_pos=[];
for k=1:length(mask)
    if mask(k)==1
        mask_pos=[mask_pos,k];
    end
end
R_reduce=R_sub(mask_pos,mask_pos);
R_reduce_inv=inv(R_reduce);
R_sub_inv=zeros(length(mask));
R_sub_inv(mask_pos,mask_pos)=R_reduce_inv;
W_opt=R_sub_inv*P;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for l=2:N_samples
% comments(['QOL  LMS  ', num2str(1/N_samples,3)]);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%get data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
time_now=(l-1)*time_step;
time_then=(l-2)*time_step;
for k=1:N_s
    signal_k=signals(k);
    if time_then<=floor(time_now/signal_k.T_data...
       & time_now > floor(time_now/signal_k.T_data)*signal_k.T_data
        data(k,1)=get_data(signal_k.constell);
    else
        data(k,1)=data(k,1-1);
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%modulation/demodulation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
signal_temp=signals(desire_signal);
w_IF=signal_temp.w_RF;
d=zeros(N_imager,1);
for K=1:N_s
    signal_K=signals(K);

    TF_signal_K=QOLTF_signal(K);
    [m,nl=size(TF_signal_K);
    d=d+TF_signal_K*data(K,1)*exp(signal_K.w_RF-w_IF)*...
        ones(n,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%save desire signal on max detector
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%if sum(mask)==1
%   if K==desire_signal
%       output.d(1)= mask'*TF_signal_K*data(K,1)*...
%           exp(signal_K.w_RF-w_IF)*ones(n,1);
%   end
%end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%estimate P
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   if K==desire_signal
%       P_est=P_est + abs(data(K,1))^2*(mask.*TF_signal_K)*...
%           exp(signal_K.w_RF-w_IF)*ones(n,1)/N_samples;
%   end

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%noise
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
noise=zeros(N_imager,1);
if noise_pwr.lens~=0
    noise_TF_temp=noise_TF.lens;
    [m,nl=size(noise_TF_temp);
    noise_temp=sqrt(noise_pwr.lens)*(randn(n,1)+j*randn(n,1))/sqrt(2);
    noise=noisel+noise_TF_temp*noise_temp;
end

if noise_pwr.detector~=0
    for K=1:length(imager)
        noise_TF_temp=noise_TF.det{K};
        [m,nl=size(noise_TF_temp);
```

```
        noise_temp=sqrt(noise_pwr.detector)*(randn(n,1)+j*randn(n,1))/sqrt(2);
        noise_QOL_temp(K,1)=noise_TF_temp*noise_temp;
    end
    noise=noise+noise_QOL_temp;
end

if noise_pwr.imager~=0
    noise_TF_temp=noise_TF.imager;
    [m,nl=size(noise_TF_temp);
    noise_temp=sqrt(noise_pwr.imager)*(randn(n,1)+j*randn(n,1))/sqrt(2);
    noise=noise+noise_TF_temp*noise_temp;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%noise on max detector
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%output.noise(1)=mask'*noise;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%LMS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
X=(d.*mask)+(noise.*mask);
y(1)=(W(:,1).*mask)'*(X.*mask);
y_opt(1)=(W_opt.*mask)'*(X.*mask);
e_1=data(desire_signal,1)-y(1);
e(1)=e_1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%estimate R
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%R_est=R_est+X*X'/N_samples;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

W(:,1+1)=W(:,1)+2*mu*e_1'*X;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end
output.data=data;
output.e=e;
output.y=y;
output.y_opt=y_opt;
output.W_opt=W_opt;
output.W=W;
output.R=R;
output.R_est=R_est;
output.P=P;
output.P_est=P_est;
```

---- W_est_SNR.m ----

```matlab
function [SNR]=W_est_SNR(output,constell,desire_signal,N_sample,TF)

%this function estimates the SNR of y in two different ways
%BER and variance from data
%N_sample is the last N_sample of y to use for the estimation

if strcmp(constell,'QPSK')
    symbol=[1,-1,j,-j];
elseif strcmp(constell,'BPSK')
    symbol=[1,-1];
elseif strcmp(constell,'carrier')
    symbol=[1,1];
end
symbol_start=ones(length(symbol));
symbol_start_opt=ones(length(symbol));

err=0;
err_opt=0;
N_y=length(output.y);
for k=(N_y-N_sample+1):N_y
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %estimate y_signal
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    y_s_est(k+N_sample-N_y)=output.W_est*output.data(desire_signal,k)*TF;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %estimate BER
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    data_guess=slicer(output.y(k),constell);
    if data_guess~=output.data(desire_signal,k)
        err=err+1;
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %estimate eye diagram mean and variance
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for l=1:length(symbol)
        if output.data(desire_signal,k)==symbol(l)
            y_symbol(l,symbol_start(l))=output.y(k);
            symbol_start(l)=symbol_start(l)+1;
        end
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    noise(k+N_sample-N_y)=output.y(k)-output.data(desire_signal,k);
```

```matlab
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%estimate noise
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y_n_est=output.y(N_y-y-N_sample+1:N_y)-y_s_est;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%calculate BER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SNR.BER=err/N_sample;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%calculate SNR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SNR.y_SNR_est=(var(y_s_est)+mean(y_s_est)*mean(y_s_est)')/...
    (var(y_n_est)+mean(y_n_est)*mean(y_n_est)');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%calculate SNR data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SNR.data=get_data_pwr(constell)/(abs(mean(noise))^2+var(noise));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%calculate y mean and variance on symbols
for l=1:length(symbol)
    temp=y_symbol(l,[1:symbol_start(l)-1]);
    SNR.y_symbol.mean(l,1)=mean(temp);
    SNR.y_symbol.var(l,1)=var(temp);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

---- weight_noise.m ----

```matlab
function [W_noise]=wieght_noise(output,N_sample,N_W)

[m,n]=size(output.W);
n=n-1;
W_var=var(output.W(:,[n-N_sample+1:n]).').';
W_noise.mean=sum(W_var)/N_W;
if N_W==1
    W_noise.var=sum(W_var'*W_var)/(N_W-1);
else
    W_noise.var=0;
```

```matlab
symbol_start=ones(length(symbol));
symbol_start_opt=ones(length(symbol));

err=0;
err_opt=0;
N_y=length(output.y);
for k=(N_y-N_sample+1):N_y
%estimate y_signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y_s_opt(k+N_sample-N_y)=output.W_opt'*output.data(desire_signal,k)*TF;
y_s_est(k+N_sample-N_y)=W.mean'*output.data(desire_signal,k)*TF;
y_s(k+N_sample-N_y)=output.W_opt'*output.data(desire_signal,k)*TF;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%estimate BER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
data_guess=slicer(output.y(k),constell);
if data_guess~=output.data(desire_signal,k)
    err=err+1;
end
data_guess=slicer(output.y_opt(k),constell);
if data_guess~=output.data(desire_signal,k)
    err_opt=err_opt+1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%estimate eye diagram mean and variance
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for l=1:length(symbol)
    if output.data(desire_signal,k)==symbol(l)
        y_symbol(l,symbol_start(l))=output.y(k);
        symbol_start(l)=symbol_start(l)+1;
        y_symbol_opt(l,symbol_start_opt(l))=output.y_opt(k);
        symbol_start_opt(l)=symbol_start_opt(l)+1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
noise(k+N_sample-N_y)=output.y(k)-output.data(desire_signal,k);
noise_opt(k+N_sample-N_y)=output.y_opt(k)-output.data(desire_signal,k);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%estimate noise
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y_n_opt=output.y_opt([N_y-N_sample+1:N_y])-y_s_opt;
y_n_est=output.y([N_y-N_sample+1:N_y])-y_s_est;
y_n=output.y([N_y-N_sample+1:N_y])-y_s;
```

```matlab
end
```

_____ win_stats.m _____

```matlab
function [stats]=win_stats(data,window)

%this function calculates the mean and variance for a windowed set of data
%out of the data vector.  The window moves from the start of the data set
%to the end of the data set changing in size at the beginning.  the last
%'window' samples for each point is used.  if data is
%a matrix, then the rows are operated on.

[m,n]=size(data);
for k=1:m
    for l=1:n
        if l-window < 0
            start=1;
        else
            start=l-window+1;
        end
        stats.mean(k,l)=mean(data(k,[start:l]));
        stats.var(k,l)=var(data(k,[start:l]));
    end
end
```

_____ y_SNR.m _____

```matlab
function [SNR]=y_SNR(output,constell,desire_signal,N_sample,TF)

%this function estimates the SNR of y in two different ways
%BER and variance from data
%N_sample is the last N_sample of y to use for the estimation

[m,n]=size(output.W);
n=n-1;
W.mean=mean(output.W(:,[n-N_sample+1:n]).').';
SNR.W_mean=W.mean;

if strcmp(constell,'QPSK')
    symbol=[1,-1,j,-j];
elseif strcmp(constell,'BPSK')
    symbol=[1,-1];
elseif strcmp(constell,'carrier')
    symbol=[1,1];
end
```

```
end

y_n=output.y([n-N_s+1:n])-y_s;

y_s_pwr=var(y_s)+mean(y_s)*mean(y_s)';
y_n_pwr=var(y_n)+mean(y_n)*mean(y_n)';

y_SNR=y_s_pwr/y_n_pwr;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%calculate BER
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SNR.BER_opt=err_opt/N_sample;
SNR.BER=err/N_sample;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%calculate SNR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SNR.y_SNR=(var(y_s)+mean(y_s)*mean(y_s)')/(var(y_n)+mean(y_n)*mean(y_n)');
SNR.y_SNR_opt=(var(y_s_opt)+mean(y_s_opt)*mean(y_s_opt)')/...
    (var(y_n_opt)+mean(y_n_opt)*mean(y_n_opt)');
SNR.y_SNR_est=(var(y_s_est)+mean(y_s_est)*mean(y_s_est)')/...
    (var(y_n_est)+mean(y_n_est)*mean(y_n_est)');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%calculate SNR data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SNR.data=get_data_pwr(constell)/(abs(mean(noise))^2+var(noise));
SNR.data_opt=get_data_pwr(constell)/(abs(mean(noise_opt))^2+var(noise_opt));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%calculate y mean and variance on symbols
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for l=1:length(symbol)
    temp=y_symbol(1,[1:symbol_start(l)-1]);
    SNR.y_symbol.mean(1,1)=mean(temp);
    SNR.y_symbol.var(1,1)=var(temp);
    temp=y_symbol_opt(1,[1:symbol_start_opt(l)-1]);
    SNR.y_symbol_opt.mean(1,1)=mean(temp);
    SNR.y_symbol_opt.var(1,1)=var(temp);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

_____ y_SNR_wiener.m _____

function [y_SNR]=y_SNR_wiener(output,TF,desire_signal,N_s)

[m,n]=size(output.W);
n=n-1;
W.mean=mean(output.W(:,[n-N_s+1:n]).').';

for k=n-N_s+1:n
    y_s(-n+N_s+k)=W.mean'*TF*output.data(desire_signal,k);
```